



TUGAS AKHIR - KS141501

**RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST
PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESSIA
MENGUNAKAN SUPPORT VECTOR MACHINE (SVM)**

***DEVELOPING AN APPLICATION FOR POST CLASSIFICATION
ON SOCIAL MEDIA OF INDONESIA LOCAL GOVERNMENTS
USING SUPPORT VECTOR MACHINE (SVM)***

**GUNTUR KONDANG PRAKOSO
NRP 0521144 0000 153**

**Dosen Pembimbing:
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA MENGGUNAKAN *SUPPORT VECTOR MACHINE (SVM)*

GUNTUR KONDANG PRAKOSO
NRP 0521144 0000 153

Dosen Pembimbing:
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

Halaman ini sengaja dikosong



ITS
Institut
Teknologi
Sepuluh Nopember

UNDEGRADUATE THESIS - KS141501

DEVELOPING AN APPLICATION FOR POST CLASSIFICATION ON SOCIAL MEDIA OF INDONESIA LOCAL GOVERNMENTS USING SUPPORT VECTOR MACHINE (SVM)

GUNTUR KONDANG PRAKOSO
NRP 0521144 0000 153

Supervisor:

Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D

**Departement of Information System
Faculty of Information Communication and Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

Halaman sengaja dikosongkan

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI UNTUK
KLASIFIKASI POST PADA SOSIAL MEDIA
PEMERINTAH DAERAH DI INDONESIA
MENGUNAKAN SUPPORT VECTOR MACHINE
(SVM)**

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

GUNTUR KONDANG PRAKOSO

NRP. 052 1144 0000 153

Surabaya, 16 Juli 2018

**KEPALA
DEPARTEMEN SISTEM INFORMASI**

Dr. Ir. Aris Tjahyanto, M.Kom.
NIP 19650310 199102 1 001

Halaman sengaja dikosongkan

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA MENGUNAKAN SUPPORT VECTOR MACHINE (SVM)

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

GUNTUR KONDANG PRAKOSO

NRP. 052 1144 0000 153

Disetujui Tim Penguji: Tanggal Ujian: 6 Juli 2018
Periode Wisuda: September 2018

Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D

(Pembimbing I)

Faizal Johan Atletiko, S.Kom, M.T

(Penguji I)

Renny Pradina Kusumawardani, S.T, M.T, SCJP

(Penguji II)

Halaman ini sengaja dikosongkan

RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)

Nama Mahasiswa : Guntur Kondang Prakoso
NRP : 0521144 0000 153
Departemen : Sistem Informasi FTIK-ITS
Dosen Pembimbing : Nur Aini Rakhmawati,
S.Kom, M.Sc.Eng, Ph.D

ABSTRAK

Seiring berkembangnya penggunaan internet di Indonesia, semakin banyak juga pengguna internet sebagai sarana media komunikasi. Dari sebuah hasil survey dapat diketahui bahwa pada tahun 2017 lebih dari 50% penduduk Indonesia telah terhubung ke internet. Banyaknya penduduk Indonesia yang menggunakan internet juga mendorong layanan serta komunikasi pemerintah menuju media berbasis internet. Pemerintah daerah membuat layanan electronic government atau lebih dikenal dengan sebutan e-government. Salah satu contoh e-government yang diterapkan pemerintah daerah adalah melalui media website serta media sosial seperti Facebook, Twitter dan Youtube. Pemerintah sendiri masih belum ada klasifikasi ataupun kategorisasi terhadap post yang dilakukan. Hal ini menyebabkan kurangnya informasi akan post seperti apa yang mendapat sedikit atau banyak tanggapan dari masyarakat. Sedikit banyaknya tanggapan sendiri menjadi salah satu pengukuran bagi keberhasilan penyampaian informasi kepada masyarakat. Dari permasalahan tersebut diperlukan solusi untuk melakukan pengkategorian agar diketahui kategori post pada media sosial pemerintah daerah di Indonesia, dengan cara melakukan klasifikasi terhadap post media sosial milik pemerintah daerah. Hal ini dilakukan dengan menggunakan konsep crawling yang berfungsi untuk

mengumpulkan data dan metode SVM untuk mengklasifikasikan kategori postingan. Pengklasifikasian ini dilakukan dengan menggunakan bantuan Apache Kafka sebagai data pipeline dan Apache Spark untuk membantu proses pembelajaran mesin. Klasifikasi kategori dibagi menjadi 6 kategori yang dimana selanjutnya dilakukan pembuatan model berdasarkan data media sosial pemerintah daerah. Kategori yang digunakan adalah edukasi warga, informasi layanan, informasi peristiwa, promosi daerah, permintaan opini informasi dan pemberitahuan pemeliharaan. Dari pemodelan tersebut didapatkan hasil 78% dengan parameter maksimal pengulangan 10 dan parameter reguler lambda 0.03. Hasil dari model ini kemudian diimplementasikan secara streaming sehingga data postingan selanjutnya dapat diklasifikasikan secara otomatis dan ditampilkan secara visual kepada pengguna.

Kata Kunci: Pemerintah Daerah, Crawling, Media Sosial, Support Vector Machine

RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)

Nama Mahasiswa : Guntur Kondang Prakoso
NRP : 0521144 0000 153
Departemen : Sistem Informasi FTIK-ITS
Dosen Pembimbing : Nur Aini Rakhmawati,
S.Kom, M.Sc.Eng, Ph.D

ABSTRACT

Along with the development of internet usage in Indonesia, more and more also internet users as medium of communication media. From a survey, the result can be seen that in 2017 more than 50% of Indonesia's population has been connected to the internet. A large number of Indonesians who use the internet also encourages governmental services and communication to Internet-based media. Local governments make electronic government services or better known as e-government. One example of e-government applied by local government is through media websites and social media like Facebook, Twitter, and Youtube. The government itself there is still no classification or categorization of the post made. This leads to a lack of information on the post as to what has received little or much public response. Few responses themselves become one of the measurements for successful delivery of information to the public. From the problem, it is a necessary solution to categorize to know the category of post on social media of local government in Indonesia, by way of classification to social media post owned by local government. This is done by using the concept of crawling that serves to collect data and SVM methods to classify the category of posts. This classification is done using the help of Apache Kafka as pipeline data and Apache Spark to help machine learning process. Classification

of categories is divided into 6 categories which in turn are modeled based on social media data of local government. The categories used are citizens education, service information, event information, regional promotion, information opinion requests and maintenance notices. From the modeling obtained 78% results with maximum parameter repetition 10 and lambda 0.03 regular parameters. The results of this model are then implemented in streaming so that the next posting data can be classified automatically and displayed visually to the user.

Keywords: Local Government, Crawling, Social Media, Support Vector Machine

KATA PENGANTAR

Puji syukur penulis haturkan kehadiran Allah SWT yang telah memberikan anugerah dan tuntunan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir dengan judul “RANCANG BANGUN APLIKASI UNTUK KLASIFIKASI POST PADA SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM)” sebagai salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Penyusunan tugas akhir ini senantiasa mendapatkan dukungan dari berbagai pihak baik dalam bentuk doa, motivasi, semangat, kritik, saran dan berbagai bantuan lainnya. Untuk itu, secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Allah SWT yang telah memberikan hidayah serta atas ijin-Nya pula saya mampu mengerjakan tugas akhir ini hingga selesai.
2. Segenap keluarga besar penulis, terutama orang tua, yang selalu senantiasa mendoakan, memberikan motivasi, dan kebutuhan materil maupun non-materil sehingga penulis mampu untuk menyelesaikan pendidikan S1 ini dengan baik.
3. Bapak Dr. Ir. Aris Tjahyanto, M.Kom., selaku Ketua Jurusan Sistem Informasi ITS, Bapak Nisfu Asrul Sani, S.Kom, M.Sc selaku KaProdi S1 Sistem Informasi ITS serta seluruh dosen pengajar beserta staf dan karyawan di Jurusan Sistem Informasi, FTIF ITS Surabaya selama penulis menjalani kuliah.
4. Ibu Nur Aini Rakhmawati, S.Kom., M.Sc., Eng. Ph.D selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan mendukung dengan memberikan ilmu, petunjuk, dan motivasi dalam penyelesaian Tugas Akhir dan selama menjalankan masa perkuliahan penulis.

5. Bapak Faizal Johan Atletiko, S.Kom., M.T dan Ibu Renny Pradina, S.T, M.T selaku dosen penguji yang telah memberikan kritik, saran, dan masukan yang dapat menyempurnakan Tugas Akhir ini.
6. Saudara-saudara EGOVBENCH yang telah menjadikan karya Tugas Akhir ini menjadi nyata, tanpa mereka mungkin saya sudah beralih profesi ke ternak lele.
7. Keluarga besar E-Home yang telah memberikan pelajaran, hitam dan putihnya dunia kampus ITS.
8. Komplotan Begundalhood yang telah memberikan motivasi, pelajaran hidup dan juga candaan gelap. Tak henti-hentinya berbagi kebahagiaan dan keluhan bersama.
9. Sesama pejuang UKM Cinta Lab dan seluruh stakeholder ADDI, yang telah memberikan sarana prasarana serta ilmu akademis.
10. Keluarga OSR yang telah memberikan dukungan emosional dan doa selama ini.
11. Rahma yang memberi semangat dan motivasi untuk mengakhiri tugas akhir ini.

Terima kasih atas segala bantuan, dukungan, serta doanya. Semoga Allah SWT senantiasa melimpahkan anugerah serta membalas kebaikan yang telah diberikan kepada penulis. Penulis menyadari bahwa masih terdapat kekurangan dalam penyusunan tugas akhir ini, oleh karena itu penulis mengharapkan saran dan kritik yang membangun demi kebaikan penulis dan tugas akhir ini. Akhir kata, penulis berharap bahwa tugas akhir ini dapat memberikan kebermanfaatan.

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
LEMBAR PERSETUJUAN.....	ix
ABSTRAK.....	xi
ABSTRAK.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR TABLE.....	xxi
DAFTAR GAMBAR	xxiii
DAFTAR KODE.....	xxiv
1. BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Relevansi.....	4
2. BAB II TINJAUAN PUSTAKA.....	5
2.1 Studi Sebelumnya	5
2.2 Dasar Teori	9
2.2.1 Sosial Media pada Pemerintah	9
2.2.2 Klasifikasi Post	11
2.2.3 <i>Web Crawler</i>	14
2.2.4 <i>Kafka</i>	14
2.2.5 <i>Support Vector Machine</i>	15
2.2.6 Kernel Trick.....	16
3. BAB III METODOLOGI.....	19
3.1 Tahapan Pelaksanaan Tugas Akhir	19
3.2 Uraian Metodologi	19
3.2.1 Studi Literatur	20
3.2.2 Akuisisi Data	20

3.2.3	Pra-pengolahan Corpus	20
3.2.4	Klasifikasi <i>Support Vector Machine</i>	21
3.2.5	Pengujian	21
3.2.6	Visualisasi	22
3.2.7	Penyusunan buku tugas akhir	24
4.	Bab IV PERANCANGAN	25
4.1	Pengambilan Data	25
4.2	Desain Database	30
4.2.1	Database Pemda	30
4.2.2	Database Facebook	31
4.2.3	Database Twitter Tweet	32
4.2.4	Database Youtube Post	33
4.3	Pembuatan Model	34
4.3.1	Crawling Data	34
4.3.2	Pelabelan data	35
4.3.3	Pra-proses Data	35
4.3.4	Pembuatan Model	36
4.3.5	Validasi Model	36
4.4	<i>Stream Classification</i> dan Visualisasi	37
4.4.1	Streaming Data	37
4.4.2	Pra-proses Data	37
4.4.3	Klasifikasi Data	37
4.4.4	Desain antarmuka aplikasi visualisasi	38
5.	BAB V IMPLEMENTASI	41
5.1	Lingkungan Implementasi	41
5.2	<i>Crawling</i> Data Sosial Media Pemerintah Daerah	41
5.2.1	Pengambilan Data Akun	42
5.2.2	<i>Crawling</i> Post Facebook	45
5.2.3	<i>Crawling</i> Post Twitter	52
5.2.4	<i>Crawling</i> Post Youtube	55
5.3	Pemilahan Data <i>Crawling</i>	61
5.4	Labelling Data Post	61
5.5	Pra-processing Data	62
5.6	Pembuatan dan Pengujian Model	65
5.7	<i>Streaming</i> Data Post	69

5.8 Streaming Spark.....	77
5.9 Klasifikasi Data.....	80
5.10 Visualisasi.....	87
6. Bab VI HASIL DAN PEMBAHASAN	94
6.1 Analisa Hasil Pemodelan	94
6.1.1 Pemuatan Data	94
6.1.2 Pembentukan Model Klasifikasi dan Validasi...	95
6.2 Analisa Data.....	99
7. BAB VII KESIMPULAN DAN SARAN	109
7.1 Kesimpulan	110
7.2 Saran	111
DAFTAR PUSTAKA	114

Halaman ini sengaja dikosongkan

DAFTAR TABLE

Table 2.1-1 Klasifikasi pada Facebook usage in government – a cases study of information content	7
Table 2.2-1 Kategori	11
Table 2.2-2 Contoh Kategori Post	12
Table 4.1-1 Jumlah Akun Media Sosial	25
Table 4.1-2 Collection Facebook	26
Table 4.1-3 Collection Twitter	27
Table 4.1-4 Collection Youtube	28
Table 4.2-1 Database Pemda	30
Table 4.2-2 Collection Facebook Post	31
Table 4.2-3 Collection Post Type Result Facebook	31
Table 4.2-4 Collection Twitter Tweet	32
Table 4.2-5 Collection Tweet Type Result Twitter	32
Table 4.2-6 Collection Post Youtube	33
Table 5.1-1 Spesifikasi Perangkat Keras	41
Table 5.1-2 Library Python	41
Table 5.1-3 Spesifikasi Perangkat Lunak	41
Table 5.2-1 Contoh Data Google Spreadsheet	44
Table 5.4-1 Hasil Labelling	61
Table 5.5-1 Hasil Cleaning	64
Table 5.5-2 Hasil Tokenizing	64
Table 5.5-3 Hasil Stopword Remover	65
Table 5.6-3 Nilai Per Kategori	69
Table 6.1-1 Pembagian Data Berdasarkan Sosial Media	94
Table 6.1-2 Hasil Pelabelan Data	94
Table 6.1-3 Hasil Tahapan Preprocessing	95
Table 6.1-4 Hasil Evaluasi Presisi, Recall, F-Measure	96
Table 6.1-5 Hasil Tiap Media Sosial	98
Table 6.1-6 Hasil Evaluasi Keseluruhan	98
Table 6.1-7 Kata Kunci Setiap Label	98
Table 6.2-1 Persebaran Data Twitter Tipe Text	100
Table 6.2-2 Persebaran Data Twitter Tiap Kategori	100
Table 6.2-3 Persebaran Data FB Tipe Link	101
Table 6.2-4 Persebaran Data FB Tipe Photo	102
Table 6.2-5 Persebaran Data FB Tipe Status	103
Table 6.2-6 Persebaran Data FB Tipe Video	104

Table 6.2-7 Persebaran Data FB Tiap Kategori	105
Table 6.2-8 Persebaran Data Youtube Tiap Kategori	106

DAFTAR GAMBAR

Gambar 2.2-1 Diagram Kerja Kafka	15
Gambar 2.2-2 Konsep SVM	16
Gambar 2.2-3 Contoh Hasil SVM	16
Gambar 2.2-4 Contoh Konsep Kernel	17
Gambar 3.1-1 Alur Metodologi	19
Gambar 3.2-1 Arsitektur Sistem	23
Gambar 4.1-1 Tree Diagram Facebook JSON	27
Gambar 4.1-2 Tree Diagram Twitter JSON	28
Gambar 4.1-3 Tree Diagram Youtube JSON	29
Gambar 4.3-1 Alur Proses <i>Crawling</i>	34
Gambar 4.3-2 Arsitektur Kafka	35
Gambar 4.3-3 Alur Pra-Proses Data	36
Gambar 4.3-4 Alur Pembuatan Model	36
Gambar 4.4-1 Alur <i>Streaming</i>	37
Gambar 4.4-2 Alur Pemrosesan Data	38
Gambar 4.4-3 Rancangan Visualisasi Status Harian	38
Gambar 4.4-4 Rancangan Visualisasi Klasifikasi Post	39
Gambar 5.9-1 Klasifikasi Data Stream	86
Gambar 5.10-1 Visualisasi Statistik Harian	89
Gambar 5.10-2 Klasifikasi Post	91
Gambar 5.10-3 Pergerakan Post	92
Gambar 5.10-4 Jenis Pos Facebook	93
Gambar 5.10-5 Jenis Pos Twitter	93
Gambar 6.1-1 Hasil Evaluasi	96
Gambar 6.1-2 Trend regParam	97
Gambar 6.1-3 Trend maxIter	97
Gambar 6.2-1 Persebaran Data Twitter Tipe Text	100
Gambar 6.2-2 Persebaran Twitter Tiap Kategori	101
Gambar 6.2-3 Persebaran Data FB Tipe Link	102
Gambar 6.2-4 Persebaran Data FB Tipe Photo	103
Gambar 6.2-5 Persebaran Data FB Tipe Status	104
Gambar 6.2-6 Persebaran Data FB Tipe Video	105
Gambar 6.2-7 Persebaran Data FB Tiap Kategori	106
Gambar 6.2-8 Persebaran Data Youtube Tiap Kategori	107
Gambar 6.2-9 Statistik Tipe Video	107
Gambar 6.2-10 Statistik Tipe Photo	108

Gambar 6.2-11 Statistik Tipe Teks 108

DAFTAR KODE

Kode 5.2-1 Fungsi Mengambil Akun dari Google Spreadsheets	43
Kode 5.2-2 Penyusunan Dictionary Untuk Post Facebook	49
Kode 5.2-3 Fungsi <i>Crawl</i> Facebook Setiap Pemda	50
Kode 5.2-4 <i>Crawl</i> Tweet Twitter	52
Kode 5.2-5 Fungsi JSON Dictionary	53
Kode 5.2-6 Fungsi <i>Crawl</i> Video Youtube	57
Kode 5.2-7 Fungsi <i>Crawl</i> Setiap Pemda.....	59
Kode 5.3-1 Pemilahan Data.....	61
Kode 5.6-1 Membagi Dataset.....	65
Kode 5.6-2 Fungsi Pengujian Tiap Kategori	68
Kode 5.7-1 Start server zookeeper	70
Kode 5.7-2 Pembuatan topik	70
Kode 5.7-3 Inisiasi Produser dan Konsumer	71
Kode 5.7-4 Parser Post	72
Kode 5.7-5 Mengirimkan Data ke Kafka	74
Kode 5.7-6 Proses Data dari <i>Crawler</i>	74
Kode 5.7-7 Twitter Steam Listener	75
Kode 5.7-8 Ambil JSON.....	75
Kode 5.7-9 Streaming Tweet Twitter	75
Kode 5.8-1 Inisiasi Spark Streaming Post	77
Kode 5.8-2 Parsing Data Twitter	78
Kode 5.9-1 Fungsi Preprocessing	80
Kode 5.9-2 Load SVM Model.....	81
Kode 5.9-3 Pengecekan Data	81
Kode 5.9-4 Proses Klasifikasi Stream	82
Kode 5.9-5 Parsing Pandas ke JSON.....	83
Kode 5.9-6 Inisiasi MongoDB	83
Kode 5.9-7 Insert Hasil Klasifikasi ke Database	84
Kode 5.9-8 Upsert Query pada MongoDB	85
Kode 5.10-1 Query Harian Post	87
Kode 5.10-2 Query Visualisasi Klasifikasi Post.....	90

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Bab pendahuluan ini menguraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1 Latar Belakang Masalah

Seiring dengan berkembangnya penggunaan Internet di Indonesia, semakin banyak juga pengguna Indonesia yang menggunakan Internet sebagai media komunikasi. Berdasar survei yang dilakukan oleh WeAreSocial dan Hootsuite pada tahun 2018 dalam laporan Global Digital 2018 terdapat 132,7 juta penduduk Indonesia yang sudah menggunakan Internet dari total penduduk Indonesia 265,4 juta penduduk. Dari survei tersebut juga diketahui perkembangan pengguna sosial media aktif sebanyak 23% dihitung dari Januari 2017. Hal ini menyebabkan pengguna sosial media aktif meningkat mencapai angka 130 juta pengguna[1]. Dari hasil survei tersebut diketahui bahwa sekitar 50% dari penduduk Indonesia terhubung ke Internet dan dari pengguna internet tersebut sebanyak 97,9% merupakan pengguna sosial media aktif. Banyaknya penduduk Indonesia yang menggunakan Internet serta menjadi pengguna media sosial aktif mendorong layanan serta komunikasi pemerintah melalui media berbasis online.

Pemerintah daerah membuat layanan electronic government atau e-government. Salah satu contoh e-governement adalah melalui media website pemerintah daerah dan sosial media seperti Facebook, Twitter dan Youtube. Menurut Permenpan no. 83 tahun 2012 sosial media membentuk dan mendukung cara baru dalam berkomunikasi serta menyediakan cara yang lebih cepat dan tepat untuk menyebarkan informasi[2]. Sosial

media pun menjadi salah satu jembatan yang menghubungkan pemerintah dengan masyarakat.

Sosial media sebagai media komunikasi pemerintah terhadap masyarakat dituntut untuk dapat menyampaikan kebijakan serta program pemerintah dengan jelas dan sederhana. Hal ini dilakukan agar masyarakat mudah untuk memahami informasi yang disampaikan[3]. Dari penggunaan sosial media oleh pemerintah daerah masih belum ada klasifikasi atau kategorisasi terhadap postingan yang dilakukan di setiap media sosial milik pemerintah. Hal ini menyebabkan kurangnya informasi akan post seperti apa yang mendapat banyak atau sedikit tanggapan dari masyarakat. Selain itu tujuan penyampaian informasi dan memberikan pemahaman akan menjadi hilang dengan sedikitnya tanggapan dari masyarakat.

Dari permasalahan diatas diperlukan pengkategorian untuk dapat diketahui kategori postingan pada media sosial pemerintah daerah di Indonesia, dengan cara melakukan analisis terhadap postingan media sosial Pemda, menggunakan metode Support Vector Machine atau yang dapat disebut SVM. Penelitian ini ditujukan untuk media sosial Facebook, Twitter dan Youtube milik seluruh Pemda di Indonesia. Setelah data postingan didapatkan dan dianalisa menggunakan SVM, kemudian dilakukan visualisasi hasil dari analisis tersebut. Melalui penelitian ini diharapkan mampu memberikan informasi melalui visualisasi serta pengkategorian postingan setiap Pemda sehingga dapat menjadi saran dan pembelajaran bagi pemerintah untuk meningkatkan serta mengelola komunikasi dengan masyarakat melalui sosial media.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang, maka rumusan permasalahan yang menjadi fokus dan akan diselesaikan dalam tugas akhir ini antara lain:

1. Bagaimana mengakuisisi data postingan Pemda di Indonesia melalui media sosial Facebook, Twitter dan Youtube?

2. Bagaimana mengkategorikan postingan Pemda dengan menggunakan SVM?
3. Bagaimana memvisualisasikan postingan Pemda?

1.3 Batasan Masalah

Batasan permasalahan dalam pengerjaan tugas akhir ini adalah:

1. Studi kasus yang digunakan pada penelitian ini adalah media sosial Facebook, Twitter dan Youtube pemerintah daerah di Indonesia.
2. Data yang digunakan pada pengerjaan Tugas Akhir ini hanya dari data post pada halaman resmi Facebook, Twitter dan Youtube.
3. Pengambilan data menggunakan *API* setiap media sosial sesuai dengan aturan dan protokol yang diberikan seperti limitasi dan sebagainya.
4. Data post yang diolah merupakan data post dengan menggunakan bahasa Indonesia, bukan bahasa daerah.
5. Klasifikasi komentar menggunakan algoritma *Support Vector Machine* digunakan hanya sebatas untuk visualisasi dan mengetahui topik post pada sosial media pemerintahan.

1.4 Tujuan Penelitian

Berdasarkan hasil perumusan masalah dan batasan masalah yang telah disebutkan sebelumnya, maka tujuan yang dicapai dari tugas akhir ini adalah untuk mengkategorikan serta melihat jenis postingan seluruh Pemda di Indonesia pada media sosial Facebook, Twitter dan Youtube guna untuk memberikan masukan kepada pemerintah daerah post seperti apa yang cocok dan mendapat banyak tanggapan oleh masyarakat.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari tugas akhir ini adalah:

1. Mempermudah pemerintah untuk mengetahui apa dan seberapa banyak topik yang dipost pada sosial media pemerintah.
2. Menyediakan data yang dapat digunakan untuk melakukan pengambilan keputusan dalam meningkatkan layanan pemerintah sesuai dengan kebutuhan di masyarakat.
3. Sebagai pembelajaran untuk penelitian di bidang *e-government* dan kaitannya dengan penggunaan algoritma *support vector machine* untuk melakukan klasifikasi topik yang dipost pada sosial media pemerintahan di Indonesia.

1.6 Relevansi

Relevansi tugas akhir ini terhadap laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI) adalah karena tugas akhir ini berkaitan dengan mata kuliah Pemrograman Berbasis Web, Analisa dan Desain Perangkat Lunak dan Konstruksi Pengembangan Perangkat Lunak, Sistem Cerdas dan Penggalan Data Analitika Bisnis

BAB II

TINJAUAN PUSTAKA

Bab ini akan membahas penelitian sebelumnya yang berhubungan dengan tugas akhir dan teori - teori yang berkaitan dengan permasalahan tugas akhir ini.

2.1 Studi Sebelumnya

Pada subbab ini dijelaskan tentang referensi penelitian yang berkaitan dengan tugas akhir. Pada bagian ini memaparkan acuan penelitian sebelumnya yang digunakan oleh penulis dalam melakukan penelitiannya.

1. Penelitian pertama berjudul “What makes local governments' online communications successful? Insights from a multi-method analysis of Facebook” oleh Sara Hofmann et al, paper ini menunjukkan penelitian mengenai pemanfaatan penggunaan media sosial oleh pemerintah daerah di Jerman yang memiliki tujuan untuk melakukan komunikasi eksternal dengan masyarakat. Penggunaan media sosial pada pemerintahan daerah ini bertujuan untuk mengatasi kekurangan dari komunikasi tradisional atau offline yang dilakukan. Pengukuran keberhasilan komunikasi dalam penelitian ini dilakukan dengan mengacu pada frekuensi serta polaritas komentar masyarakat terhadap postingan pemerintah. Hasil yang didapat dari penelitian tersebut adalah fitur multimedia seperti foto dan video berkontribusi untuk kesuksesan komunikasi yang dilakukan[4].
2. Penelitian kedua berjudul “The Public Facebook: A Case of Australian Government Facebook Pages and Participation” oleh Sultana Lubna Alam dan David Walker, paper ini menunjukkan penelitian mengenai pemanfaatan Facebook untuk menilai keterbukaan

pemerintah serta partisipasi online audiens pada enam FB Pemerintahan Australia. Penelitian ini bertujuan untuk mendapatkan pemahaman tipe form yang cocok digunakan oleh pemerintahan serta agensi mana yang akan menangani hal tersebut. Pada paper ini terdapat lima pembagian kategori utama, yaitu giving information, requesting information, positive comment, negative comment, dan miscellaneous. Metode yang digunakan adalah dengan melihat statistic dari frekuensi post, jumlah likes, pembuat konten dan feedback atau reaksi dari konten yang didapat. Hasil yang didapat dari penelitian tersebut adalah pandangan mengenai tipe komunikasi yang umum dilakukan oleh organisasi atau pemerintahan[5].

3. Penelitian ketiga berjudul “Social media in Egyptian government websites: Presence, usage, and effectiveness” oleh Abdelsalam et al, paper ini membahas tentang keberadaan, penggunaan serta efektivitas situs media sosial pemerintah Mesir. Pada tahap pertama dari penelitian ini adalah dengan menguji keberandaan aplikasi media sosial pada situs web pemerintah. Selanjutnya tahap kedua dilakukan analisis penggunaan Facebook oleh badan pemerintah Mesir. Pemilihan Facebook disebabkan karena merupakan salah satu aplikasi media sosial teratas selain Twitter dan Youtube pada situs web pemerintah Mesir[6].
4. Penelitian keempat berjudul “Twitter News Classification Using SVM” oleh Inoshika Dilrukshi et al, paper ini menunjukkan penelitian mengenai klasifikasi berita pada Twitter dengan menggunakan metode SVM. Setiap pesan yang terdapat dalam tweet diklasifikasikan ke dalam 12 grup. Metode SVM digunakan karena SVM mendukung high dimensional data. Dimana pada high dimensional data memiliki sejumlah fitur atau dimensi dalam skala besar. Selain itu SVM tidak overfitting pada data sehingga dapat

meminimalis kesalahan serta kompleksitas. SVM juga memiliki fitur untuk memunculkan global minimum. Pengukuran efektifitas atau validasi dilakukan dengan menggunakan nilai recall dan precisions[7].

5. Penelitian kelima berjudul “Facebook usage in government – a cases study of information content” oleh Monika Magnusso et al, paper ini menunjukkan penelitian mengenai penggunaan Facebook oleh pemerintahan dimana studi kasus yang diambil mengenai analisis konten informasi didalamnya. Data didapat dari status updates pemerintah dan user posting pada wall. Data yang didapat kemudian diunduh dan dicetak. Selanjutnya konten dievaluasi dan dikategorisasikan dalam proses iterative coding. Dari hasil pengolahan data tersebut didapatkan kategori akhir seperti berikut [8]:

Table 2.1-1 Klasifikasi pada Facebook usage in government – a cases study of information content

Municipality		
<i>Content category</i>	<i>Abbreviation</i>	<i>Description</i>
Educating Citizens	EC	Elucidate citizen, e.g. ‘serious games’ or by helping citizens educate each other
Marketing of Events	ME	Promoting different happenings in the municipality
Marketing of Municipality	MM	Promoting the municipality as a good place to live or work

Marketing of Services	MS	Promoting municipal services
Request Information from Citizen	RIC	Asking citizens for information or assistance
Service Maintenance Information	SMI	Informing of service break-downs and ongoing repairs
Citizens		
Citizen Marketing Events	CME	Citizen promoting events that are not directly connected to a commercial organization
Citizen Sharing Information	CSI	Sharing information, links etc. that are not marketing of a an event or business
Complaining about Municipality	CM	Citizen holding government responsible
Express Opinion	EO	Citizen stating his/her point of view in a municipal matter
Identity or Community Building	ICB	Citizen promote oneself or the municipality
Marketing of Business	MB	Citizen promoting a

		commercial organization
Praising Municipality	PM	Giving positive feedback to the municipality
Request of Existing Service	RES	Appeal for improvement in present service
Request of Future Service	RFS	Appeal for launching of non-existing service
Request for Information	RI	Asking a question to the municipality or other citizens
Report Service Breakdown	RSB	Inform the municipality of problems with their services

2.2 Dasar Teori

Berisi teori-teori yang mendukung serta berkaitan dengan tugas akhir yang sedang dikerjakan.

2.2.1 Sosial Media pada Pemerintah

Sosial media merupakan sekumpulan alat online yang digunakan untuk melakukan interaksi sosial. Berbeda dengan media komunikasi tradisional seperti buku, radio dan televisi, sosial media menyediakan komunikasi yang bisa dilakukan secara dua arah atau lebih (many-to-many). Pemanfaatan sosial media dalam ruang lingkup pemerintahan menyebabkan terbukanya peluang dalam perkembangan teknologi informasi saat ini dengan adanya partisipasi dan keterikatan pengguna, kebutuhan dalam meningkatkan informasi dan layanan, serta

peningkatan inovasi untuk mengembangkan isu publik terkini[9]. Pada Permenpan nomor 83 tahun 2012 disebutkan terjadinya perubahan pendekatan dalam berhubungan dengan media serta pemerintah senantiasa dituntut untuk meningkatkan kemampuannya dalam menghadapi tantangan dan perubahan lingkungan. Sosial media memiliki manfaat untuk menyebarluaskan informasi pemerintah, meningkatkan interaksi serta menggali aspirasi dan opini masyarakat[2]. Beberapa sosial media yang saat ini digunakan oleh pemerintah daerah adalah sebagai berikut:

2.2.1.1 Facebook

Facebook merupakan situs jejaring sosial gratis yang membolehkan pengguna untuk membuat profil, mengunggah foto dan video, mengirim pesan, bermain game, serta stream live video. Facebook juga menyediakan fitur untuk mengatur post sehingga dapat dilihat oleh publik atau hanya bisa diakses oleh sekelompok orang yang kita inginkan[10]. Sekitar 130 juta pengguna aktif setiap bulannya. Pertumbuhan platform ini mencapai 23% per Januari 2018. Facebook sendiri menduduki peringkat kedua dengan kategori sosial media teraktif berdasarkan survey yang dilakukan oleh Hootsuite per Januari 2018[11]. Facebook dalam hal ini menyediakan fitur yang mampu menyampaikan informasi serta pelayanan dalam bentuk post yang memiliki beberapa jenis variasi post. Berikut variasi post yang ada pada facebook:

- | | |
|-----------|----------|
| a. Status | d. Album |
| b. Foto | e. Note |
| c. Video | |

2.2.1.2 Twitter

Twitter merupakan layanan sosial networking microblogging gratis yang membolehkan pengguna terdaftar untuk menyiarkan post pendek yang disebut tweets. Pengguna Twitter dapat menyiarkan tweets serta mengikuti (follow) pengguna lainnya dengan menggunakan berbagai platform serta perangkat[10].

Twitter sendiri menduduki peringkat ketujuh dengan kategori sosial media teraktif berdasarkan survey yang dilakukan oleh Hootsuite per Januari 2018[11]. Twitter memiliki kesamaan dengan Facebook dalam menyediakan sarana menyampaikan informasi serta pelayanan meskipun tidak sepanjang post yang disediakan oleh Facebook. Namun Twitter sendiri memiliki kelebihanannya sendiri yang bersifat lebih ringkas. Berikut variasi post yang ada pada Twitter:

- | | |
|-----------|----------|
| a. Status | c. Foto |
| b. GIF | d. Video |

2.2.1.3 Youtube

Youtube merupakan sebuah situs video yang dimana penggunanya dapat mengunggah, menyimpan serta membagikan video melalui situs tersebut. Pengguna dapat membagikan video dengan melalui berbagai platform dengan membagikan tautan (link) atau dengan menambahkan (embedding) pada kode HTML[10]. Youtube sendiri menduduki peringkat pertama dengan kategori sosial media teraktif berdasarkan survey yang dilakukan oleh Hootsuite per Januari 2018[11]. Youtube merupakan sarana khusus bagi pemerintah daerah untuk menyampaikan informasi serta pelayanan melalui media video.

2.2.2 Klasifikasi Post

Klasifikasi post merupakan klasifikasi konten pada post yang dilakukan oleh pemda. Klasifikasi ini diambil berdasarkan pada paper penelitian kelima[8]. Dimana dataset yang digunakan pada penelitian tersebut adalah data post pada Facebook milik pemerintahan Swedia.

Table 2.2-1 Kategori

Kategori Konten	Singkatan	Deskripsi
Edukasi Warga	EW	Peningkatan pengetahuan kepada warga, membantu warga untuk meningkatkan wawasan

Informasi Peristiwa	IP	Pemberitahuan atau pengumuman mengenai suatu peristiwa pada daerah tersebut
Promosi Daerah	PD	Mempromosikan bahwa daerah tersebut menjadi tempat yang nyaman untuk ditinggali atau bekerja, kelebihan daerah tersebut
Informasi Layanan	IL	Pemberitahuan atau informasi mengenai layanan yang diberikan atau yang dimiliki oleh daerah tersebut
Permintaan Informasi / Opini Kepada Warga	PIO	Permintaan pendapat kepada warga, permintaan opini kepada warga terhadap suatu hal yang bersangkutan terhadap daerah tersebut atau mengenai hal lainnya
Pemberitahuan Pemeliharaan (<i>maintenance</i>)	PP	Pemberitahuan pemeliharaan kerusakan, terjadinya kerusakan atau malfungsi suatu layanan yang ada pada daerah tersebut.

Table 2.2-2 Contoh Kategori Post

Kategori Konten	Contoh Post
Edukasi Warga	“Berhenti yang katanya ngomong modern kalau

	<p>minum-minuman keras! berhenti kalau yang katanya modern itu narkoba!" Ujar Walikota pagi tadi di SMP 8 Surabaya. Selengkapnya klik tautan berikut yaa dulur"</p> <p>@sapawargakotasurabaya</p>
Informasi Acara	<p>"Besok ada pasar murah lagi nih lur! Ada dimana saja? Monggo disimak infonya 😊😁</p> <p>Jangan sampai kelewatan ya, borong semua mumpung murah 😊😁😁 😊😁"</p> <p>@sapawargakotasurabaya</p>
Informasi Daerah	<p>"Dulur, tiga Sekolah Dasar dan tiga Sekolah Menengah Pertama meraih juara Sekolah Adiwiyata tingkat Kota Surabaya 2018. Siapa saja juaranya? Monggo cek di tautan berikut. Siapa tahu ada sekolahmu, dulur 😊"</p> <p>@sapawargakotasurabaya</p>
Informasi Layanan	<p>"Dulur, Taman Bungkul sudah resmi punya alat parkir meter lho. Bagaimana berita selengkapnya? Yuk baca di tautan berikut 😊"</p>

	@sapawargakotasurabaya
Permintaan Informasi/Opini Kepada Warga	-
Pemberitahuan Pemeliharaan (<i>maintenance</i>)	“Info Gangguan PDAM” @sapawargakotasurabaya

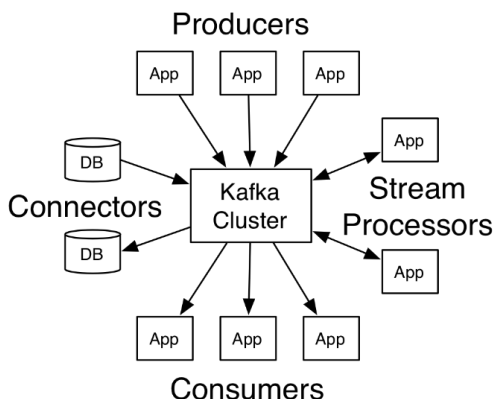
2.2.3 *Web Crawler*

Web Crawler atau juga dapat dikenal sebagai robot laba-laba. *Web crawler* berbentuk sebagai program script dimana memiliki metode tertentu sehingga dapat melakukan proses pemindaian ke halaman-halaman web untuk membuat indeks dari data yang menjadi tujuan pencarian[12]. Secara umum *web crawler* dibagi menjadi dua bagian utama, yaitu bagianb offline dan bagian online. Bagian offline dieksekusi secara periodik oleh mesin pencari, dan pada proses tersebut, *crawler* mengunduh beberapa bagian tertentu dari web yang kemudian disatukan sehingga menjadi *searchable index*. Pada bagian online, dieksekusi setiap kali terdapat permintaan pengguna ymag dieksekusi, dan menggunakan index untuk memilih beberapa kandidat dokumen yang sudah diurutkan menurut keinginan yang diharapkan pengguna.

2.2.4 *Kafka*

Kafka merupakan sebuah aplikasi *open-source* yang digunakan untuk membuat suatu data *pipelines* yang bekerja secara real-time serta digunakan untuk *streaming apps*. *Kafka* memperbolehkan untuk melakukan pengambilan data, menyimpan dan memberikan data yang diinginkan sesuai dengan kebutuhan. *Kafka* dapat mengumpulkan data lebih dari satu sumber yang kemudian disalurkan ke dalam beberapa sumber pengolahan. Hal ini bisa dilakukan tanpa perlu terlebih dahulu melakukan registrasi untuk meminta data[13]. Oleh karena itu *Kafka* dapat dikatakan juga sebagai *pull-based data aggregator* seperti halnya *RabbitMQ*. Pada tugas akhir ini

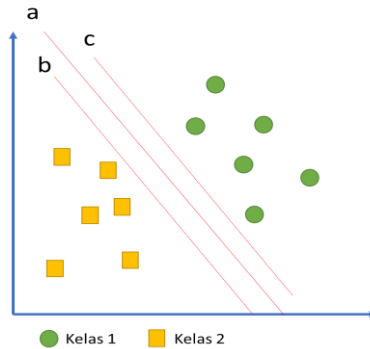
Kafka menjadi pintu saluran atau pipa yang menghubungkan data-data pada sosial media untuk dapat diambil dan diolah.



Gambar 2.2-1 Diagram Kerja Kafka

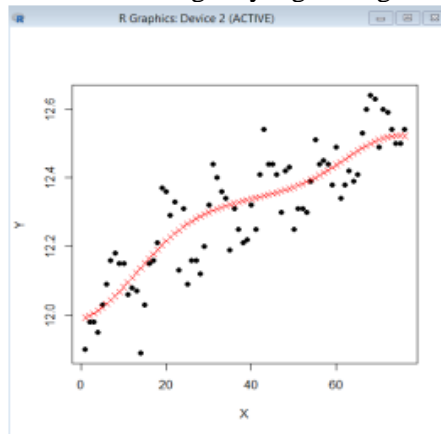
2.2.5 *Support Vector Machine*

SVM merupakan salah satu teknik machine learning yang digunakan untuk melakukan prediksi dimana metode ini masuk ke dalam kelas supervised learning. SVM pertama kali diperkenalkan oleh Vapnik pada tahun 1992. Metode SVM dapat digunakan untuk melakukan prediksi baik dalam kasus klasifikasi maupun regresi[12]. Metode SVM memiliki keuntungan dimana dapat menangani high dimensional data. Dimana pada high dimensional data memiliki sejumlah fitur atau dimensi dalam skala besar. Selain itu SVM tidak overfitting pada data sehingga dapat meminimalis kesalahan serta kompleksitas. SVM juga memiliki fitur untuk memunculkan global minimum[7]. Prinsip dasar SVM adalah linear classifier dimana SVM mencari hyperplane pemisah antar kelas. Pencarian hyperplane ini didasarkan pada margin yang terdapat antar kelas. Margin sendiri adalah jarak antara hyperplane dimana semakin besar jarak yang dimiliki maka semakin baik sehingga hyperplane tersebut yang digunakan.



Gambar 2.2-2 Konsep SVM

Pada gambar hyperplane (a) yang dipilih disebabkan karena margin antara kelas (1) dan kelas (2) paling besar dan seimbang dibandingkan hyperplane (b) dan (c). Hal ini dilakukan untuk menghindari dan meminimalisir kesalahan penempatan kelas. Berikut merupakan contoh penerapan SVM pada R dengan data jumlah wisatawan mancanegara yang datang ke Indonesia;

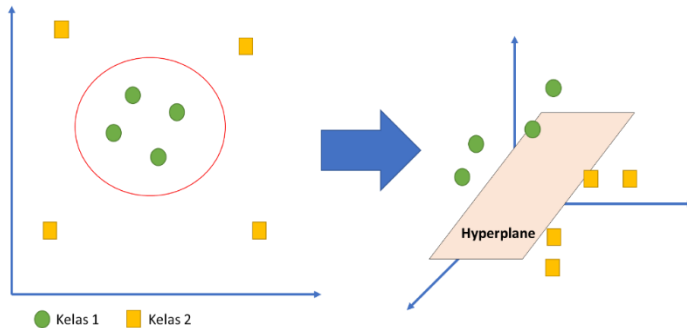


Gambar 2.2-3 Contoh Hasil SVM

2.2.6 Kernel Trick

Kernel merupakan salah satu fungsi yang digunakan dalam SVM. Kernel memungkinkan untuk mentransformasi input ruang berdimensi rendah menjadi ruang berdimensi tinggi (*high*

dimensional space) sehingga permasalahan pemisahan kelas dapat dilakukan. Berdasarkan Cover, jika suatu transformasi bersifat non-linier dan dimensi dari ruang fitur (*feature space*) cukup tinggi, maka data pada ruang masukan dapat dipetakan ke ruang fitur yang baru, dimana pola-pola tersebut pada probabilitas tinggi dapat dipisahkan secara linier. Dari pernyataan tersebut kernel dapat memungkinkan penyelesaian permasalahan pemisahan non-linier pada SVM.



Gambar 2.2-4 Contoh Konsep Kernel

Pada gambar tersebut dapat dijabarkan data pada kelas 1 dan kelas 2 bila diletakkan pada ruang masukan berdimensi dua maka pemisahan untuk menemukan *hyperplane* secara linier tidak dimungkinkan. Sehingga perlunya penambahan dimensi agar dapat ditemukan *hyperplane* secara linier.

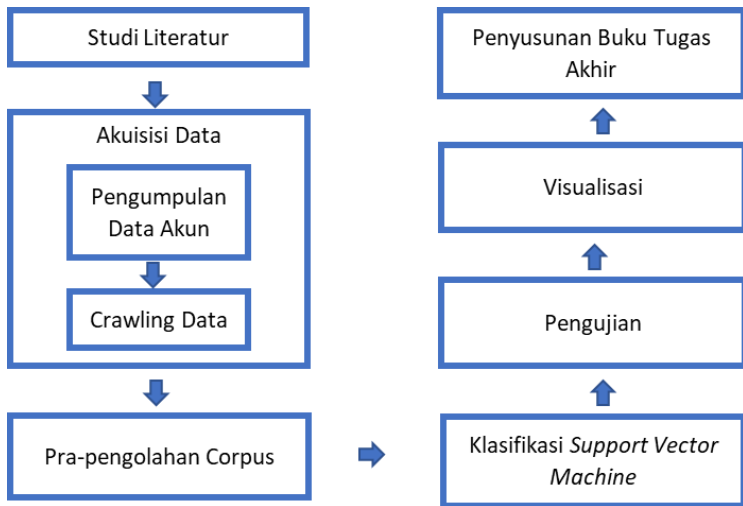
Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Bab ini menjelaskan tentang metodologi yang akan digunakan dalam penyusunan tugas akhir. Metodologi akan digunakan sebagai panduan dalam penyusunan tugas akhir agar terarah dan sistematis.

3.1 Tahapan Pelaksanaan Tugas Akhir

Pada subbab ini akan menjelaskan mengenai metodologi dalam pelaksanaan tugas akhir. Metodologi ini dapat dilihat pada



Gambar 3.1-1 Alur Metodologi

3.2 Uraian Metodologi

Pada bagian ini akan dijelaskan secara lebih rinci masing-masing tahapan yang dilakukan untuk penyelesaian tugas akhir ini.

3.2.1 Studi Literatur

Tahap studi literatur ini dilakukan untuk memahami konsep, metode, dan teknologi sesuai bahasan dan permasalahan sehingga dapat memberikan solusi mengenai permasalahan yang akan digunakan dalam penyusunan tugas akhir. Adapun literatur yang digunakan dalam penelitian ini adalah terkait klasifikasi, *Support Vector Machine*.

3.2.2 Akuisisi Data

Pada tahap ini dilakukan penggalian data yang kemudian akan diteliti. Pada tahap ini terdapat dua sub tahap didalamnya yaitu pengumpulan akun yang kemudian dilanjutkan dengan crawling data.

3.2.2.1 Pengumpulan data akun

Tahap mempersiapkan data terdiri dari beberapa aktivitas yakni mengumpulkan data akun Facebook, Twitter dan Youtube, pengumpulan postingan akun Facebook, Twitter dan Youtube terpilih yang kemudian dilakukan pemrosesan data rangkaian tahapan ini dilakukan untuk mempersiapkan dokumen yang akan dianalisis menggunakan metode SVM.

3.2.2.2 Crawling Data

Adapun untuk pengambilan data akun yang diambil adalah data dari akun milik seluruh pemerintah daerah di Indonesia yang telah dikumpulkan. Pengambilan data dilakukan dengan memanfaatkan crawler.

3.2.3 Pra-pengolahan Corpus

Pada tahap ini dilakukan pengolahan dokumen yang sudah terkumpul, pengolahan dilakukan dalam beberapa tahap yaitu beberapa langkah berikut.

- a. *Lowercase*; dimana teks perlu dibentuk menjadi *lowercase* dengan tujuan agar kata yang sama namun

berbeda secara penulisan huruf kapital dan tidak, tidak dianggap berbeda.

- b. *Tokenization*; pada proses ini terdapat dua langkah, langkah pertama adalah *stopwords*, dimana diseleksi kata umum (*common words*) yang biasa muncul dalam jumlah besar dan dianggap tidak memiliki makna. Langkah selanjutnya adalah *stemming*, langkah ini digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut. Langkah ini bekerja dengan menghilangkan imbuhan, baik yang terdiri dari awalan, sisipan, akhiran dan kombinasi dari awalan dan akhiran pada setiap kata.

3.2.4 Klasifikasi *Support Vector Machine*

Pada tahap ini data yang telah terkumpul dibagi menjadi dua bagian yaitu data training dan data testing. Pelabelan pada data training dilakukan dengan cara manual dengan menggunakan bantuan Microsoft Excel. Hal ini dilakukan pada data yang selanjutnya akan digunakan untuk klasifikasi menggunakan SVM. Kemudian dilakukan klasifikasi dengan menggunakan SVM, klasifikasi didasarkan sesuai dengan 6 topik konten yang dipilih. Hal ini dilakukan dengan membangun pemodelan SVM sebelumnya.

3.2.5 Pengujian

Pada tahap ini dilakukan pengujian serta penilaian hasil dari klasifikasi model SVM. Pengujian klasifikasi digunakan dengan mengukur *precisions*, dimana presisi digunakan untuk mengukur ketepatan prediksi pengklasifikasi pada kelas tertentu. Berikut merupakan rumus dari presisi;

$$Precision = \frac{TP}{TP + FP}$$

Equation 3.2-1 Precision

Selanjutnya terdapat pengukuran *recall*, dimana pengukuran ini digunakan untuk memperkirakan seberapa banyak instans dari

suatu kelas dapat diprediksi secara benar. Berikut merupakan rumu *recall*;

$$Recall = \frac{TP}{TP + FN}$$

Equation 3.2-2 Recall

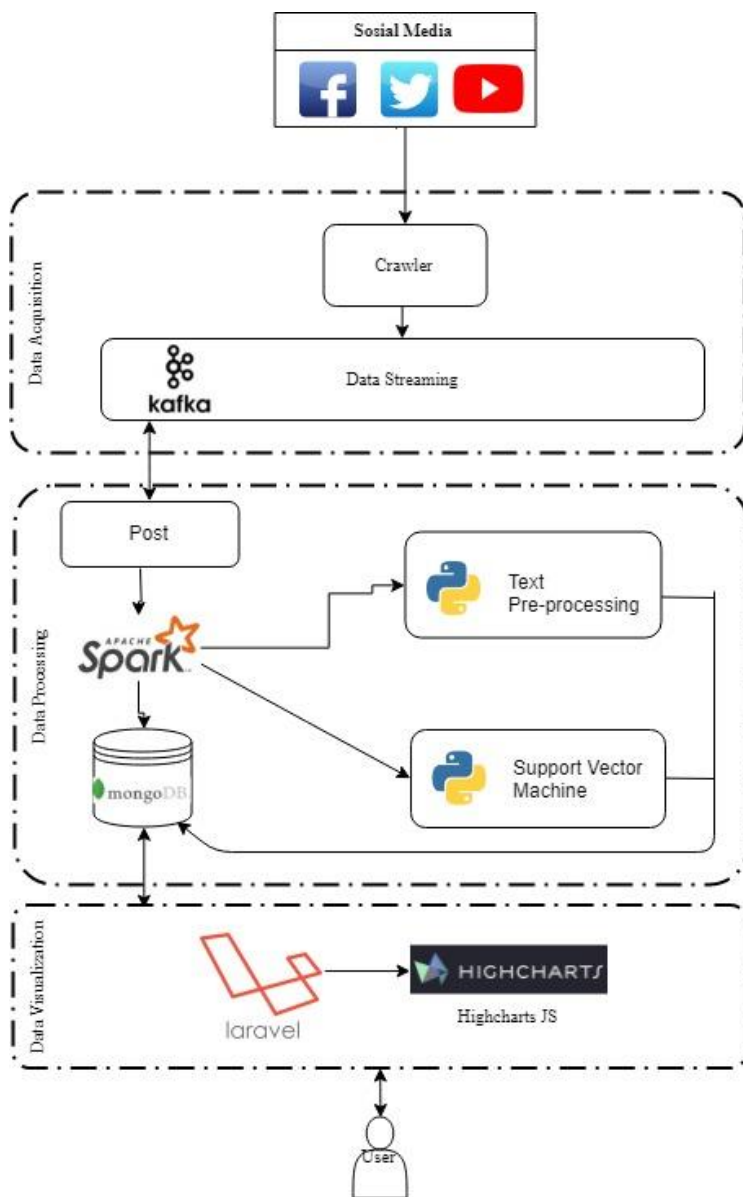
Setelah didapat kedua nilai dari pengukuran sebelumnya selanjutnya dilakukan pengukuran dengan menggunakan *F-measures*, dimana pengukuran ini digunakan untuk melakukan penggabungan kedua pengukuran menjadi satu perhitungan. Berikut merupakan dari rumus dari perhitungan harmonik (*F-measures*);

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 3.2-3 F-Measures

3.2.6 Visualisasi

Pada tahap ini hasil dari pengolahan SVM divisualisasikan. Visualisasi ini digunakan untuk memperlihatkan kecenderungan post pada setiap daerah atau wilayah serta post seperti apa yang paling banyak dilakukan oleh setiap pemda. Selain itu visualisasi ini juga digunakan untuk memperlihatkan seberapa aktif pemda memberikan interaksi kepada masyarakat melalui media sosial. Visualisasi menggunakan gabungan beberapa chart, salah satunya map chart, bubble chart, bar chart serta line chart. Berikut merupakan susunan arsitektur sistem :



Gambar 3.2-1 Arsitektur Sistem

3.2.7 Penyusunan buku tugas akhir

Pada tahap ini seluruh proses dan hasil dari pengerjaan keseluruhan tugas akhir didokumentasikan dan disusun menjadi sebuah buku. Selain digunakan untuk dokumentasi proses dan hasil dari penelitian, buku tugas akhir ini juga dapat digunakan sebagai bahan referensi untuk penelitian selanjutnya.

Bab IV

PERANCANGAN

Pada bab ini dijelaskan terkait alur perancangan terkait beberapa hal yang diperlukan dalam pembuatan aplikasi sesuai dengan alur yang dijelaskan pada bab 3.

4.1 Pengambilan Data

Dalam melakukan pengambilan data, objek yang diambil merupakan post pada sosial media Facebook, Twitter, dan Youtube pada akun resmi pemerintah kota pada masing-masing daerah. Proses pengambilan data dilakukan dengan dua cara yaitu *crawling* dan *streaming*. Data yang diambil dengan cara *crawling* digunakan sebagai bahan dalam pembuatan model sementara data yang diambil dengan cara *streaming* digunakan sebagai data yang nantinya akan dikategorikan dengan menggunakan model yang telah dibuat. Pengambilan data terlebih dahulu mengambil data postingan pada ke tiga sosial media dengan menggunakan API yang tersedia pada masing-masing platform sosial media. Setelah data didapatkan selanjutnya dilakukan *preprocessing* data dan disimpan ke dalam database menggunakan MongoDB. Berikut merupakan jumlah akun sosial media resmi pemerintah kota per tanggal 22 Mei 2018:

Table 4.1-1 Jumlah Akun Media Sosial

Sosial Media	Jumlah Akun Resmi
Facebook	149
Twitter	202
Youtube	113
Total Akun	464

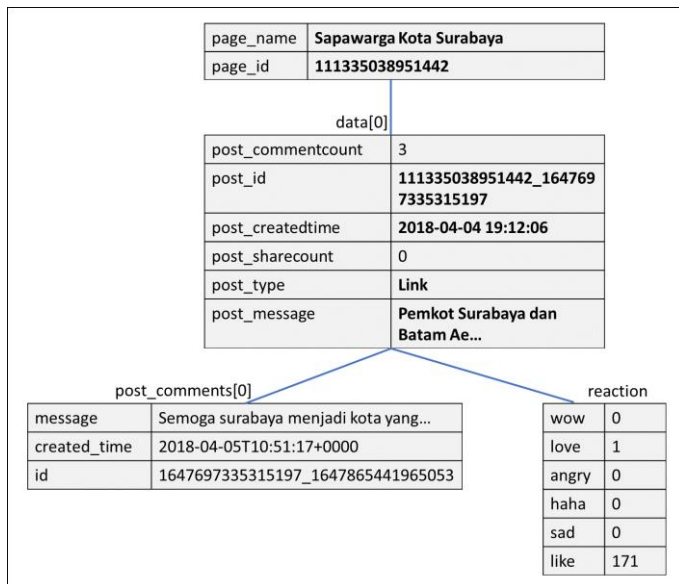
Proses pengambilan data dilakukan dengan dua cara yaitu *streaming* dan *crawling*. Data yang diambil dengan cara *crawling* digunakan sebagai bahan pembuatan model, sedangkan data *streaming* digunakan sebagai data yang akan diklasifikasikan. Pengambilan data dilakukan melalui API yang disediakan pada masing-masing media sosial. Adapun atribut

data yang diambil pada media sosial Facebook dapat dilihat pada Tabel berikut.

Table 4.1-2 Collection Facebook

Field	Keterangan	Contoh
page_id	Nomor ID Facebook pemda	111335038951442
page_name	Nama Facebook pemda	Sapawarga Kota Surabaya
posts.post_id	Nomor ID postingan	111335038951442_1647697335315197
posts.post_message	Isi postingan	Pemkot Surabaya dan Batam Aero Technic (BAT) Siap Jalin Kerjasama.
posts.post_type	Tipe post yang dikirim	link
posts.post_sharecount	Jumlah tautan dibagikan	0
posts.post_comment.id	Nomor ID komentar	1647697335315197_1647865441965053
posts.post_comment.message	Isi komentar	Semoga surabaya menjadi kota yang lebih baik lagi
posts.post_comment.created_time	Waktu pengiriman komentar	2018-04-05T10:51:17+0000
posts.post_commentcount	Jumlah komentar	3
posts.reaction.like	Jumlah reaction like	171
posts.reaction.love	Jumlah reaction love	1

posts.reaction.wow	Jumlah reaction wow	0
posts.reaction.haha	Jumlah reaction haha	0
posts.reaction.sad	Jumlah reaction sad	0
posts.reaction.angry	Jumlah reaction angry	0
posts.created_time	Waktu pengiriman post	2018-04-04 19:12:06



Gambar 4.1-1 Tree Diagram Facebook JSON

Untuk atribut atribut data yang diambil pada media sosial Twitter dapat dilihat pada Tabel berikut

Table 4.1-3 Collection Twitter

Field	Keterangan	Contoh
id	Nomor ID tweet	982281671130017793

text	Isi tweet	@pln_123 @SapawargaSby Lampu jalan kota di JL. TAMBANG BOYO mati... Kenapa yaa?? Seremm tau ga sih.. Tolong segera di nyalakan...
screenname	Nama akun	aljoki1990
favorite_count	Jumlah favorit pada tweet	0
retweet_count	Jumlah retweet pada tweet	0
created_at	Waktu pengiriman post	Fri Apr 06 15:39:34 +0000 2018
type	Tipe media tweet	

id	982281671130017793
text	@pln_123 @SapawargaSby Lampu jal...
screenname	aljoki1990
favorite_count	0
retweet_count	0
created_at	Fri Apr 06 15:39:34 +0000 2018
type	

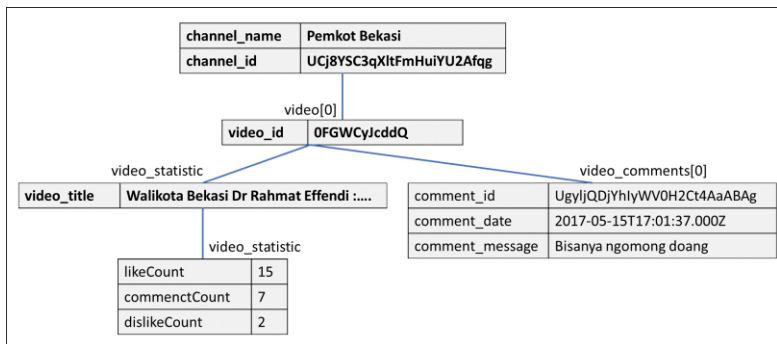
Gambar 4.1-2 Tree Diagram Twitter JSON

Untuk atribut atribut data yang diambil pada media sosial Youtube dapat dilihat pada Tabel berikut

Table 4.1-4 Collection Youtube

Field	Keterangan	Contoh
channel_id	Nomor ID channel	UCj8YSC3qXltFm HuiYU2Afqg
channel_name	Nama Youtube pemda	Pemkot Bekasi

video.video_id	Nomor ID video	0FGWCyJcddQ
video.video_info.video_title	Judul video	Walikota Bekasi Dr Rahmat Effendi : Explore Potensi Curug Parigi untuk Wujudkan #BekasiHijau
video.video_info.video_statistics.likecount	Jumlah like video	15
video.video_info.video_statistics.commentcount	Jumlah kometar video	7
video.video_info.video_statistics.dislikecount	Jumlah dislike video	2
video.video_comments.comment_id	Nomor ID komentar	UgyljQDjYhlyWV0H2Ct4AaABAg
video.video_comments.comment_date	Waktu pengiriman komentar	2017-05-15T17:01:37.000Z
video.video_comments.comment_message	Isi komentar	Bisanya ngomong doang



Gambar 4.1-3 Tree Diagram Youtube JSON

Diagram tree pada setiap JSON dibuat berdasarkan pembagian setiap pemda yang kemudian dilanjutkan dengan pembagian media sosial setiap pemda. Setelah dari pembagian media sosial selanjutnya masuk ke dalam post dan tipe post pada tiap-tiap media sosial. Hal ini dilakukan untuk mempermudah pelacakan post yang dimiliki setiap pemda pada keseluruhan media sosial.

4.2 Desain Database

Desain database dilakukan untuk memetakan penyimpanan data hasil klasifikasi serta data *stream* dari media sosial Facebook, Twitter dan Youtube.

4.2.1 Database Pemda

Database pemda berfungsi untuk menyimpan akun dari media sosial milik pemda, dimana nantinya masing-masing akun tersebut akan digunakan untuk melakukan *streaming* melalui API masing-masing media sosial.

Table 4.2-1 Database Pemda

Field	Tipe Data	Keterangan
_id	Integer	Nomor ID Pemda
name	String	Nama Pemerintah Daerah
Facebook_resmi	String	Nama akun Facebook resmi
Facebook_influencer	String	Nama akun Facebook influencer
Twitter_resmi	String	Nama akun Twitter resmi
Twitter_influencer	String	Nama akun Twitter influencer
Youtube_resmi	String	Nama akun Youtube resmi
Youtube_influencer	String	Nama akun Youtube influencer

4.2.2 Database Facebook

Database ini berfungsi untuk menyimpan hasil dari klasifikasi post dan juga tipe post pada Facebook akun resmi milik pemda. Hal ini diperlukan untuk selanjutnya data dapat digunakan membentuk visualisasi.

Table 4.2-2 Collection Facebook Post

Field		Tipe Data	Keterangan
_id		String	ID post Facebook
page_id		String	Nama halaman akun
post_comment Count		Integer	Jumlah komentar pada post
post_createdDate		String	Waktu pengiriman post
post_message		String	Isi dari post
post_reactions	wow	Integer	Jumlah masing-masing reaction pada post
	sad	Integer	
	haha	Integer	
	like	Integer	
	angry	Integer	
love		Integer	
post_shareCount		Integer	Jumlah share pada post
post_type		String	Jenis tipe dari post
class		String	Hasil tipe klasifikasi

Table 4.2-3 Collection Post Type Result Facebook

Field	Tipe Data	Keterangan
_id	String	ID objek yang tergenerate dari mongoDB
post_type	String	Jenis tipe dari post
result_createdDate	String	Tanggal hasil dibuat

result	statist c	postCo unt	Integer	Jumlah suatu tipe pada keseluruhan post Facebook
--------	--------------	---------------	---------	--

Pada post tabel 4.2.2-1 tersebut *field* yang tidak digunakan pada penelitian ini adalah post_commentCount, post_reactions dan post_shareCount. Tabel 4.2.2-2 merupakan potongan dari keseluruhan database post type result Facebook dimana potongan tersebut yang berhubungan dengan tugas akhir ini.

4.2.3 Database Twitter Tweet

Database ini berfungsi untuk menyimpan hasil dari klasifikasi post dan juga tipe tweet pada Twitter akun resmi milik pemda. Hal ini diperlukan untuk selanjutnya data dapat digunakan membentuk visualisasi.

Table 4.2-4 Collection Twitter Tweet

Field	Tipe Data	Keterangan
_id	String	ID tweet Twitter
account_id	String	Nama akun
tweet_createdDate	String	Waktu tweet dibuat
tweet_favoriteCount	Integer	Jumlah favorite pada tweet tersebut
tweet_message	String	Isi dari tweet tersebut
tweet_replyCount	Integer	Jumlah reply pada tweet tersebut
tweet_retweetCount	Integer	Jumlah retweet dari tweet tersebut
tweet_type	String	Jenis dari tweet tersebut
Class	String	Hasil tipe klasifikasi

Table 4.2-5 Collection Tweet Type Result Twitter

Field	Tipe Data	Keterangan
-------	-----------	------------

_id			String	ID objek yang tergenerate dari mongoDB
tweet_type			String	Jenis tipe dari post
account_followerCount			Integer	Tanggal hasil dibuat
result	statistic	tweetCount	Integer	Jumlah suatu tipe pada keseluruhan tweet Twitter

Pada post tabel 4.2.3-1 tersebut *field* yang tidak digunakan pada penelitian ini adalah tweet_favoriteCount, tweet_replyCount dan tweet_retweetCount. Tabel 4.2.3-2 merupakan potongan dari keseluruhan database tweet type result Twitter dimana potongan tersebut yang berhubungan dengan tugas akhir ini.

4.2.4 Database Youtube Post

Database ini berfungsi untuk menyimpan hasil dari klasifikasi post dan juga tipe post pada Facebook akun resmi milik pemda. Hal ini diperlukan untuk selanjutnya data dapat digunakan membentuk visualisasi.

Table 4.2-6 Collection Post Youtube

Field	Type Data	Keterangan
_id	String	ID video Youtube
channel_id	String	Nama channel pemda
video_commentCount	Integer	Jumlah komentar pada video
video_createdDate	String	Waktu video diupload
video_dislikeCount	Integer	Jumlah dislike pada video
video_likeCount	Integer	Jumlah like pada video
video_title	String	Judul dari video tersebut
video_viewCount	Integer	Jumlah view pada video
class	String	Hasil tipe klasifikasi

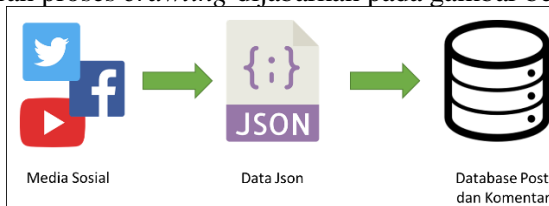
Pada post tabel 4.2.4-1 tersebut *field* yang tidak digunakan pada penelitian ini adalah video_commentCount, video_dislikeCount, video_likeCount dan video_viewCount.

4.3 Pembuatan Model

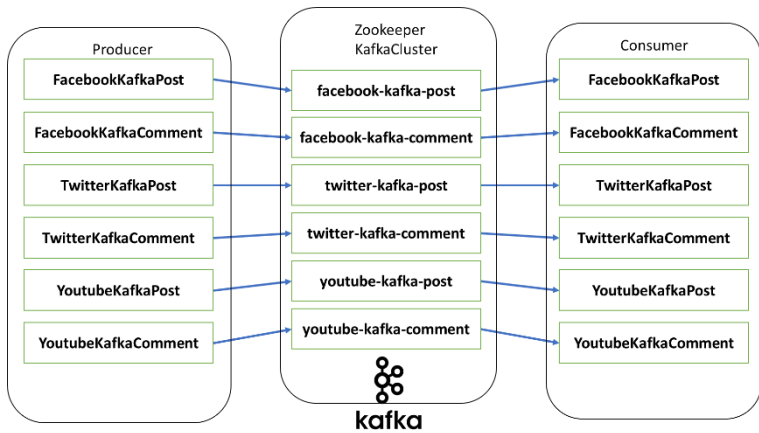
Pembuatan model dalam penelitian ini bertujuan untuk membuat model kategorisasi yang nantinya akan digunakan dalam implementasi ke dalam aplikasi. Tahapan ini terbagi menjadi 4 bagian yaitu crawling data, pelabelan data, pra-proses data, dan pembuatan model.

4.3.1 Crawling Data

Crawling data merupakan tahapan pengambilan data postingan dan komentar pada akun sosial media resmi dan influencer pemerintah daerah. Tahapan crawling ini menggunakan python dan API di masing-masing sosial media untuk mendapatkan JSON yang berisikan data postingan dan komentar. Adapun jumlah data yang didapatkan melalui proses crawling berjumlah XXXX data untuk periode XXXX 2018 hingga XXXX 2018. Gambaran proses *crawling* dijabarkan pada gambar berikut :



Gambar 4.3-1 Alur Proses *Crawling*



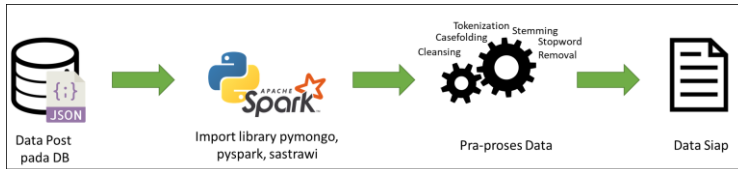
Gambar 4.3-2 Arsitektur Kafka

4.3.2 Pelabelan data

Setelah data diambil dengan menggunakan crawling, data akan dipisahkan berdasarkan jenisnya yaitu postingan. Pada penelitian ini, data yang digunakan adalah data post. Data post selanjutnya akan diberikan pelabelan sebagai tahapan dalam pembuatan model. Pelabelan dilakukan berdasarkan kategori yang dijabarkan sebelumnya serta dilakukan secara manual.

4.3.3 Pra-proses Data

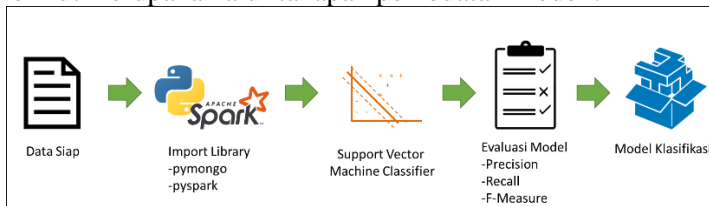
Tahapan pra-proses data mencakup beberapa langkah pengerjaan yaitu pembersihan data dari tag atau karakter-karakter tertentu, pengubahan data menjadi huruf kecil casefolding, stemming, stopword removal, serta tokenization. Stemming dilakukan dengan menggunakan Stemmer Sastrawi[14]. Stopword removal dilakukan berdasarkan penelitian yang dilakukan oleh Fadilah Z. Tala[15]. Data yang dibutuhkan adalah data yang telah diberi label pada tahap sebelumnya. Berikut merupakan gambaran pra-proses data :



Gambar 4.3-3 Alur Pra-Proses Data

4.3.4 Pembuatan Model

Pada tahapan ini, dilakukan pembuatan model klasifikasi dengan menggunakan metode SVM yang dimana menggunakan kernel linear yang telah disediakan oleh library Apache Spark *machine learning* yaitu `pyspark LinearSVC`. Pada pembuatan model ini digunakan dua parameter yang menjadi acuan yaitu maksimum iterasi dan *regularization* parameter yang berfungsi untuk menghindari misklasifikasi pada data train sekaligus untuk menentukan *soft-margin* pada model. Selanjutnya data yang didapatkan akan dipecah menjadi dua bagian yaitu 70% dan 30%. Data sebesar 70% akan digunakan sebagai data training dan 30% akan digunakan sebagai data testing. Data training digunakan untuk melatih pengetahuan terhadap model dan data testing digunakan untuk menguji pengetahuan model. Berikut merupakan alur tahapan pembuatan model :



Gambar 4.3-4 Alur Pembuatan Model

4.3.5 Validasi Model

Pada tahapan ini dilakukan validasi model untuk memastikan bahwa model yang sudah dibuat sesuai. Validasi model dilakukan dengan cara menganalisis perhitungan dari 3 perhitungan yaitu sebagai berikut.

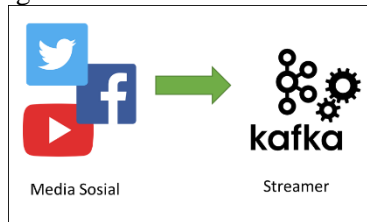
1. Hasil persentase *precision* yang baik pada model
2. Hasil persentase *recall* yang baik pada model
3. Hasil persentase *f-measure* yang baik pada model

4.4 *Stream Classification dan Visualisasi*

Metodologi implementasi dalam penelitian ini merupakan tahapan yang bertujuan untuk menyelesaikan penelitian dengan komputasi secara otomatis. Tahapan ini terbagi menjadi 4 tahapan yaitu

4.4.1 *Streaming Data*

Streaming data merupakan tahapan pengambilan data postingan dan komentar pada akun sosial media resmi dan influencer pemerintah daerah secara langsung. Tahapan streaming ini menggunakan python dan API di masing-masing sosial media untuk mendapatkan JSON yang berisikan data postingan dan komentar. *Kafka* digunakan sebagai *data pipeline* yang selanjutnya data yang didapat dibagi berdasarkan *topic*. Pada penelitian ini *topic* yang disubscribe adalah *topic* yang berkaitan dengan data post. Gambaran proses *streaming* dijabarkan pada gambar berikut :



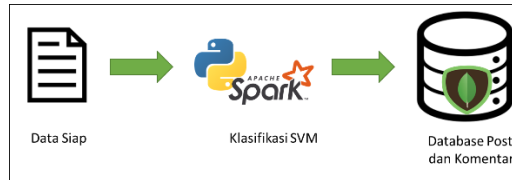
Gambar 4.4-1 Alur *Streaming*

4.4.2 *Pra-proses Data*

Tahapan pra-proses data mencakup beberapa langkah pengerjaan yaitu pembersihan data dari tag atau karakter-karakter tertentu, pengubahan data menjadi huruf kecil casefolding, stemming, stopword removal, serta tokenization. Library yang digunakan adalah library Sastrawi[14].

4.4.3 *Klasifikasi Data*

Data yang sudah siap selanjutnya dilakukan proses klasifikasi menggunakan *support vector machine* dengan mengacu pada model yang telah dibuat sebelumnya. Hal ini dilakukan melalui Python dan Apache Spark. Data yang telah diklasifikasikan selanjutnya disimpan dalam database untuk dapat divisualkan.



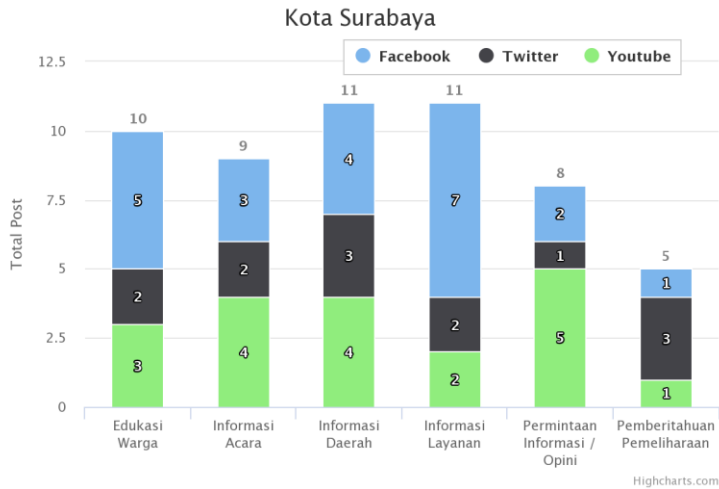
Gambar 4.4-2 Alur Pemrosesan Data

4.4.4 Desain antarmuka aplikasi visualisasi

Data yang sudah didapatkan dari hasil klasifikasi kemudian divisualisasikan ke dalam beberapa kategori yang ada. Visualisasi yang digunakan pada penelitian ini menggunakan bahasa PHP, dimana akan ditampilkan dalam bentuk dashboard kepada user. Rancangan dashboard yang akan ditampilkan dari analisa klasifikasi berdasarkan post tiap masing-masing daerah yang dipilih. Pada Gambar 4.3.3 menjelaskan tentang jumlah data post setiap media sosial yang berhasil didapatkan. Pada Gambar 4.3.4 menampilkan rancangan tampilan diagram klasifikasi post per pemda, dimana akan ditampilkan ke dalam aplikasi E-Govbench.

Daily Stats	
Total Post Facebook	21
Total Post Twitter	50
Total Post Youtube	21
Rata-rata post per Pemda	21
Jumlah Post Terklasifikasi	21

Gambar 4.4-3 Rancangan Visualisasi Status Harian



Gambar 4.4-4 Rancangan Visualisasi Klasifikasi Post

Halaman ini sengaja dikosongkan

BAB V

IMPLEMENTASI

Pada bab ini akan dijelaskan terkait proses implementasi pada perangkat lunak yang dirancang. Pada bagian ini akan dijelaskan bagaimana lingkungan implementasi, pembuatan fitur-fitur aplikasi dan proses pengujian aplikasi.

5.1 Lingkungan Implementasi

Pada bagian ini dibahas terkait kriteria perangkat pengujian dalam implementasi. Berikut spesifikasi perangkat yang digunakan dalam implementasi.

Table 5.1-1 Spesifikasi Perangkat Keras

No.	Hardware	Spesifikasi
1	Jenis	ASUS K46CM
2	Processor	Intel Core i5-3317U
3	RAM	8 GB
4	Harddisk	500GB HDD

Table 5.1-2 Library Python

No.	Library	Spesifikasi
1	pandas 0.23.0	Dataframe
2	Sastrawi 1.1.0	Stemming
3	pyspark 2.2.1	Pra-proses & Klasifikasi (Linear SVC)
4	re & string	Pra-proses
5	json	JSON Flattened

Table 5.1-3 Spesifikasi Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1	Apache 2.4.33 & PHP 7.2.5	Webserver
2	Python 3.6 32-bit	Pemrosesan Data
3	Jupyter Notebook	Python IDE

4	MongoDB 3.6	Database Server
5	Google Chrome 67.0.3396.79	Web Browser
6	Sublime Text 3 & Atom 1.27.2	Editor

5.2 *Crawling* Data Sosial Media Pemerintah Daerah

Crawling data pada sosial media pemda digunakan untuk mengambil data post yang ada pada setiap sosial media sehingga dapat digunakan untuk membentuk model klasifikasi. Proses *crawling* dilakukan dengan menggunakan Facebook Graph API untuk Facebook, Twitter API dan tweepy untuk Twitter dan Google API untuk Youtube.

5.2.1 Pengambilan Data Akun

Untuk melakukan *crawling* dari ketiga media sosial diambil data akun dari setiap pemerintahan daerah untuk mendapatkan data yang diinginkan. Daftar akun dari setiap pemda diambil dari google spreadsheet yang berisi kumpulan data akun setiap pemda. Untuk mengambil data tersebut dilakukan dengan kode berikut.

Kode 5.2-1 Fungsi Mengambil Akun dari Google Spreadsheets

```
import gspread
from oauth2client.service_account import
ServiceAccountCredentials
import pprint

scope = ['https://spreadsheets.google.com/feeds',
'https://www.googleapis.com/auth/drive']

creds =
ServiceAccountCredentials.from_json_keyfile_name('cl
ient_secret.json', scope)
client = gspread.authorize(creds)

pp = pprint.PrettyPrinter()

def scrapingcolumn(col_number, fromrow=0,
torow=None, baru=False):
    # situspemda = sheet.get_all_records()

    if baru:
        sheet = client.open('situs.csv').sheet4
    else:
        sheet = client.open('situs.csv').sheet1

    result = sheet.col_values(col_number)

    return result[fromrow:torow]
```

Table 5.2-1 Contoh Data Google Spreadsheet

No. Urut				Facebook	Twitter	YouTube	Facebook Influencer	Twitter Influencer	YouTube Influencer	Twitter ID Resmi
1	PROVINSINAD	http://www.acehprov.go.id	PROV	326019957540137	acehprov		acehinfo	TweetAneukAceh		1584625914
2	KAB. ACEH SELATAN	http://www.acehselatankab.go.id/	KAB				116815498393415			
3	KAB. ACEH TENGGARA	http://acehtenggarakab.go.id/	KAB							
4	KAB. ACEH TIMUR	http://acehtimurkab.go.id/index.php	KAB					AcehTimurNAD		

5.2.2 *Crawling* Post Facebook

Pada proses *crawling* post Facebook yang dilakukan adalah mengambil data dengan menggunakan Graph API v2.12 yang disediakan oleh Facebook dimana diambil beberapa *fields* yang dibutuhkan. *Fields id* dan *fields name* digunakan untuk menampilkan nama atau identitas dari akun Facebook pemda tersebut. *Fields post* digunakan untuk melihat deskripsi dari suatu post dimana didalamnya diambil *fields* lain yang sesuai dengan kebutuhan. *Fields* tersebut adalah *fields created_time* digunakan untuk mengetahui waktu post pada akun tersebut dibuat, *fields message* digunakan untuk mengambil isi post pada setiap akun pemda, *fields type* digunakan untuk mengetahui tipe post yang dikirim.

Kode 5.2-2 Fungsi Request

```
def request_until_succeed(url):
    req = Request(url)
    success = False
    while success is False:
        try:
            response = urlopen(req)
            if response.getcode() == 200:
                success = True
                # print('.', end='', flush=True)
        except Exception as e:
            print('\n', e)
            time.sleep(1)

            print("Error for URL {}: {}".format(url,
datetime.datetime.now()))
            print("Retrying...")

    return response.read()
```

Kode 5.2.-2 menunjukkan fungsi yang berfungsi untuk melakukan pemanggilan URL Facebook melalui Graph API. Pemanggilan akan dilakukan berulang hingga berhasil.

Kode 5.2-3 Fungsi *Crawl* Post Facebook

```

def scrapeFacebookPageFeedStatus(page_id,
access_token, since_date, until_date, number,
pemandaname, type):
    has_next_page = True
    num_processed = 0
    scrape_starttime = datetime.datetime.now()
    after = ''
    base = "https://graph.facebook.com/v2.12"
    node = "{}/{}".format(page_id)
    # parameters =
    "?limit={}&access_token={}".format(100,
access_token)
    parameters =
    "?access_token={}&fields=id,name,fan_count,posts.lim
it({})".format(access_token, 100)
    since = ".since({})".format(since_date) if
since_date \
        is not '' else ''
    until = ".until({})".format(until_date) if
until_date \
        is not '' else ''

    print("\n(~~~~~(づ。●_●。)づ
FACEBOOK SCRAPER (●_●*)~~~~~)")
    print("Scraping No.{} :\n Pemda : {}\n Type
: {}\n Page ID : {}\n at : {}".format(number,
pemandaname, type, page_id, scrape_starttime))

    complete_dict = {}
    complete_dict['page_id_name'] = page_id
    complete_dict['post'] = []

```



```

        base_url = base + node + parameters + since
+ until

        url = getFacebookPageFeedUrl(base_url)

        j_input =
json.loads(request_until_succeed(url).decode())

        complete_dict['page_id_number'] =
j_input['id']
        complete_dict['page_name'] = j_input['name']
        complete_dict['page_fanCount'] =
j_input['fan_count']

        print(' Page ID (number) : {}\n Page name :
{}\n Total fans :
{}\n'.format(complete_dict['page_id_number'],
complete_dict['page_name'],
complete_dict['page_fanCount']))

        while has_next_page:

            after = '' if after is '' else
".after({})".format(after)
            base_url = base + node + parameters +
since + until + after

            url = getFacebookPageFeedUrl(base_url)

            j_input =
json.loads(request_until_succeed(url).decode())

            if 'posts' in j_input:
                after =
j_input['posts']['paging']['cursors']['after']
                comments =
getCommentsForStatuses(base_url)
                # print(reactions)

                for status in
j_input['posts']['data']:

                    status_data =
processFacebookPageFeedStatus(status)
                    status_data['post_comments'] =
comments[status_data['post_id']]

```

```

complete_dict['post'].append(status_data)

        num_processed += 1
        if num_processed % 100 == 0:
            print("{} Statuses
Processed: {}".format
                                (num_processed,
datetime.datetime.now()))

        else:
            has_next_page = False

            print("\nDone!\n{} Statuses Processed in
{}".format(
                num_processed, datetime.datetime.now()
- scrape_starttime))

        return complete_dict

```

Kode 5.2-3 menunjukkan fungsi untuk mengambil data post dari setiap akun Facebook milik pemda. Dimana membutuhkan input `page_id` yang merupakan akun Facebook milik pemda, `since_date` dan `until_date` merupakan waktu mulai dan waktu akhir pengambilan data. Pada fungsi tersebut juga dilakukan pengambilan URL sesuai dengan URL dan *field* yang dibutuhkan. Data yang didapat kemudia disusun dalam *dictionary* tersendiri yang sesuai dengan kebutuhan. Pada kode ini juga dilakukan pemanggilan fungsi `processFacebookPageFeedStatus`, dimana fungsi ini melakukan penyusunan *dictionary* untuk post yang berhasil didapatkan. Pada kode 5.2-4 dilakukkan penyusunan dictionary untuk setiap post yang berhasil didapatkan. Fungsi tersebut mengolah dictionary untuk setiap satu status (post) yang masuk.

Kode 5.2-2 Penyusunan Dictionary Untuk Post Facebook

```

def processFacebookPageFeedStatus(status):
    serialized_dict = {}
    serialized_dict['reaction'] = {}
    serialized_dict['post_id'] = status['id']
    serialized_dict['post_type'] = status['type']
    serialized_dict['post_message'] = '' if
'message' not in status else \
        unicode_decode(status['message'])
    status_published = datetime.datetime.strptime(
        status['created_time'], '%Y-%m-
%dT%H:%M:%S+0000')
    status_published = status_published + \
        datetime.timedelta(hours=-7) # EST
    status_published = status_published.strftime(
        '%Y-%m-%d %H:%M:%S') # best time format for
spreadsheet programs
    serialized_dict['post_createdtime'] =
status_published
    serialized_dict['post_commentcount'] = 0 if
'comments' not in status else
status['comments']['summary']['total_count']
    serialized_dict['post_sharecount'] = 0 if
'shares' not in status else
status['shares']['count']
    serialized_dict['reaction']['like'] =
status['like']['summary']['total_count']
    serialized_dict['reaction']['love'] =
status['love']['summary']['total_count']
    serialized_dict['reaction']['wow'] =
status['wow']['summary']['total_count']
    serialized_dict['reaction']['haha'] =
status['haha']['summary']['total_count']
    serialized_dict['reaction']['sad'] =
status['sad']['summary']['total_count']
    serialized_dict['reaction']['angry'] =
status['angry']['summary']['total_count']
    return serialized_dict

```

Kode 5.2-3 Fungsi *Crawl* Facebook Setiap Pemda

```

def pagelists():
    start_row = 1
    namapemda = spreadsheet.scrapingcolumn(1,
start_row, None)
    idpemda = spreadsheet.scrapingcolumn(3,
start_row, None)
    idpagepemda = spreadsheet.scrapingcolumn(7,
start_row, -1)
    idpageinfluencer =
spreadsheet.scrapingcolumn(10, start_row, -1)
    since_date = "2018-05-01"
    until_date = ""

    access_token = <acces token Facebook>

    for satuanidpemda in idpemda[298:]:

        pemda_dict = {}
        pemda_dict['pemda_id'] = satuanidpemda
        pemda_dict['pemda_name'] =
namapemda[idpemda.index(satuanidpemda)]
        pemda_dict['resmi'] = {}
        pemda_dict['influencer'] = {}

        satuanidpagepemda =
idpagepemda[idpemda.index(satuanidpemda)]
        satuanidinfluencer =
idpageinfluencer[idpemda.index(satuanidpemda)]

        if satuanidpagepemda is not '':
            pemda_dict['resmi'] =
scrapeFacebookPageFeedStatus(satuanidpagepemda,
access_token, since_date, until_date, satuanidpemda,
pemda_dict['pemda_name'], 'resmi')

            elif satuanidpagepemda is '':
                pemda_dict['resmi']['post'] = []

        pemda_dict['resmi']['post'].append({'post_comments':
[]})

```

```

        if satuanidinfluencer is not '':
            pemda_dict['influencer'] =
scrapeFacebookPageFeedStatus(satuanidinfluencer,
access_token, since_date, until_date, satuanidpemda,
pemda_dict[pemda_name], 'influencer')

        elif satuanidinfluencer is '':
            pemda_dict['influencer']['post'] = []

pemda_dict['influencer']['post'].append({'post_comme
nts': []})

        if satuanidpemda is idpemda[0]:
            exported = '[' +
str(json.dumps(pemda_dict, indent=4)) + ','

            with open('Facebook.json', 'a') as file:
                file.write(exported)

        elif satuanidpemda is not idpemda[-1]:
            exported = str(json.dumps(pemda_dict,
indent=4)) + ','

            with open('Facebook.json', 'a') as file:
                file.write(exported)

        else:
            exported = str(json.dumps(pemda_dict,
indent=4)) + ']'

            with open('Facebook.json', 'a') as file:
                file.write(exported)

```

Pada kode 5.2-5 akun milik pemda dimuat dalam bentuk *list* dimana data akun tersebut didapatkan dari pengambilan data pad google spreadsheet. Rentang waktu mulai dan akhir diinisiasi pada kode ini. Selanjutnya *crawler* akan berjalan secara berulang sesuai dengan list yang berisi kaun-akun pemda. Kode ini juga melakukan pengecekan apakah pemda

memiliki akun atau tidak. Data hasil pada proses ini merupakan JSON lengkap yang berisi data dari keseluruhan akun pemda.

5.2.3 *Crawling Post Twitter*

Pada proses *crawling* Twitter yang dilakukan adalah mengambil data *tweet* pada Twitter dengan menggunakan Twitter API dimana diambil beberapa *fields* yang sesuai dengan kebutuhan. *Fields tweet* untuk mengambil isi post tersebut. *Fields tweet_type* untuk menentukan tipe dari tweet tersebut. *Fields created_at* untuk mengetahui kapan tweet tersebut dibuat. *Fields id_str* sebagai identitas dari tweet tersebut.

Kode 5.2-4 *Crawl Tweet Twitter*

```
def crawlaccounts(akun, id_pemda, nama_pemda,
complete_dict):
    num_processed = 0
    complete_dict['account_name'] = akun
    complete_dict['tweets'] = []
    try:
        for tweets in tweepy.Cursor(
            TwitterAPI.user_timeline,
            screen_name=akun,
            count=100,
            include_rts=True,
            tweet_mode='extended').items():
            json_str = json.dumps(tweets._json)
            j_results = json.loads(json_str)
            ambiljson(j_results, complete_dict,
akun)

            num_processed += 1
            if num_processed == 1:
                complete_dict['follower_count'] =
j_results['user']['followers_count']

                if num_processed % 100 == 0:
                    print("{} Tweet Processed:
{}".format(num_processed,
datetime.datetime.now()))
                    crawltweets(akun, num_processed,
complete_dict)
            except tweepy.TweepError as e:
                logging.exception("message")
```

Kode 5.2-7 merupakan fungsi yang melakukan crawl tweets untuk mengambil tweet. API yang digunakan adalah API Search dengan pengaturan extended tweet mode. Pengaturan ini berfungsi untuk mengambil keseluruhan teks tweet sehingga tidak terpotong.

Kode 5.2-5 Fungsi JSON Dictionary

```
def ambiljson(j_results, complete_dict, akun):
    if 'RT @' not in j_results['full_text']:
        complete_tweet = {}
        complete_tweet['tweet_id'] =
j_results['id_str']
        complete_tweet['tweet_message'] =
j_results['full_text']
        complete_tweet['created_at'] =
j_results['created_at']
        complete_tweet['retweet_count'] =
j_results['retweet_count']
        complete_tweet['favorite_count'] =
j_results['favorite_count']
        complete_tweet['tweet_type'] = "text"
        if 'entities' in j_results:
            if 'media' in j_results['entities']:
                complete_tweet['tweet_type'] =
j_results['entities']['media'][0]['type']
            if 'extended_entities' in j_results:
                if 'media' in
j_results['extended_entities']:
                    complete_tweet['tweet_type'] =
j_results['extended_entities']['media'][0]['type']
                complete_tweet['reply_count'] = 0
                if j_results['user']['screen_name'] == akun:

complete_dict['tweets'].append(complete_tweet)

    return complete_dict
```

Kode 5.2-8 merupakan kode yang berfungsi untuk menyusun hasil *crawl* menjadi *dictionary* yang diinginkan sehingga sesuai dengan yang dibutuhkan pada pengerjaan tugas akhir ini.

Kode 5.2-9 Fungsi Dictionary Kosong

```
def nullValues(complete_dict):
    complete_dict = {}
    complete_dict['tweets'] = []

    return complete_dict
```

Kode 5.2-9 memiliki fungsi untuk menghindari adanya error disebabkan adanya data kosong. Data kosong terjadi karena pemda tidak memiliki akun resmi tapi memiliki akun influencer dan sebaliknya.

Kode 5.2-10 Fungsi *Crawl* Tweet setiap Pemda

```
def accountlists():
    start_row = 1
    id_pemda_list = spreadsheet.scrapingcolumn(3,
start_row, None)
    nama_pemda_list = spreadsheet.scrapingcolumn(4,
start_row, None)
    account_list = spreadsheet.scrapingcolumn(8,
start_row, -1)
    account_id_list = spreadsheet.scrapingcolumn(13,
start_row, -1)
    for id_pemda in id_pemda_list[389:]:
        complete_dict = {}
        complete_dict['pemda_id'] = id_pemda
        nama_pemda =
nama_pemda_list[id_pemda_list.index(id_pemda)]
        complete_dict['pemda_name'] = nama_pemda
        account =
account_list[id_pemda_list.index(id_pemda)]
        if account is not '':
            account_id =
account_id_list[id_pemda_list.index(id_pemda)]
            complete_dict['account_id'] = account_id

            print('\n', id_pemda, '. ', nama_pemda,
':', account, '(', account_id, ') ' + '\n-----
-----')
            crawlaccounts(account, id_pemda,
nama_pemda,
                        complete_dict)
```



```

else:
    nullValues(complete_dict)
    exported = str(json.dumps(complete_dict,
indent=4))
    if id_pemda == id_pemda_list[0]:
        with open('Twitter.json', 'w') as file:
            file.write('[' + exported + ',')
    elif id_pemda == id_pemda_list[-1]:
        with open('Twitter.json', 'a') as file:
            file.write(exported + ']')
    else:
        with open('Twitter.json', 'a') as file:
            file.write(exported + ',')

```

Pada kode 5.2-10 akun milik pemda dimuat dalam bentuk *list* dimana data akun tersebut didapatkan dari pengambilan data pada google spreadsheet. Rentang waktu mulai dan akhir diinisiasi pada kode ini. Selanjutnya *crawler* akan berjalan secara berulang sesuai dengan list yang berisi kaun-akun pemda. Kode ini juga melakukan pengecekan apakah pemda memiliki akun atau tidak. Data hasil pada proses ini merupakan JSON lengkap yang berisi data dari keseluruhan akun pemda.

5.2.4 *Crawling Post Youtube*

Pada proses *crawling* Youtube yang dilakukan adalah mengambil data video pada Youtube dengan menggunakan Google API dimana terdapat beberapa tahap. Tahap pertama *crawl* dilakukan dengan berdasarkan pada fitur *search* pada Google API Youtube v3 untuk mendapatkan video id dalam suatu channel dengan menggunakan *fields id/videoId, snippet/channelTitle*. Selanjutnya menggunakan fitur *video* pada Google API Youtube v3 untuk mendapatkan judul dari setiap video pada channel melalui *fields id, snippet/title*.

Kode 5.2-11 Fungsi Mengubah Username ke Channel ID

```

def channelusernameatoid(idchannel):
    parameters = {"part": "snippet",
                  "forUsername": idchannel,
                  "key": api_key,
                  "fields":
"items(id,snippet/title)"}
    url =
"https://www.googleapis.com/Youtube/v3/channels"
    usernameconverter = {}

    page = requests.request(method='get', url=url,
params=parameters)
    j_results = json.loads(page.text)
    usernameconverter['title'] =
j_results['items'][0]['snippet']['title']
    usernameconverter['id'] =
j_results['items'][0]['id']

    return usernameconverter

```

Pada kode 5.2-11 menunjukkan fungsi untuk mengubah username id menjadi channel id sehingga dapat dilakukan *crawling*. Hal ini disebabkan API Youtube v3 menyediakan *crawling list* video pada akun dengan menggunakan channel id.

Kode 5.2-12 Fungsi Mengambil Nama Channel

```

def channelnameforfilename(idchannel):
    parameters = {"part": "snippet",
                  "id": idchannel,
                  "key": api_key,
                  "fields": "items/snippet/title"}

    url =
"https://www.googleapis.com/Youtube/v3/channels"
    page = requests.request(method='get', url=url,
params=parameters)
    j_results = json.loads(page.text)
    idconverter = {}
    if j_results['items']:
        idconverter['title'] =
j_results['items'][0]['snippet']['title']
        idconverter['id'] = idchannel
    else:

```

```

        idconverter =
channelusernameatoid(idchannel)
        return idconverter

```

Kode 5.2-12 merupakan fungsi untuk mengambil nama channel serta pengecekan apakah masih berupa username atau telah berubah menjadi channel id. Pengambilan nama channel ini dilakukan untuk membuat identitas dari akun.

Kode 5.2-6 Fungsi *Crawl Video Youtube*

```

def crawlvideo(api_key, channelId, pemdaID,
pemdaName, channelType):
    temp = channelnameforfilename(channelId)
    channelName = temp['title']
    idchannel = temp['id']
    parameters = {"part": "snippet",
                  "channelId": idchannel,
                  "maxResults": 50,
                  "publishedAfter": "2018-05-
01T00:00:00Z",
                  "key": api_key,
                  "type": "video",
                  "fields":
"items(id/videoId,snippet/channelTitle),nextPageToken"}
    url =
"https://www.googleapis.com/YouTube/v3/search"
    video_search_nextpage = True
    num_processed = 0
    scrape_starttime = datetime.datetime.now()
    print("\nScraping start for pemda no. : {} ({}
!\nYoutube Channel: {} \ntype: {} \nat:
{}\n".format(pemdaID, pemdaName, channelName,
channelType, scrape_starttime))
    complete_dict = {}
    complete_dict['channel_id'] =
parameters['channelId']
    complete_dict['video'] = []
    while video_search_nextpage:
        page = requests.request(method='get',
url=url, params=parameters)
        j_results = json.loads(page.text)
        if j_results['items']:
            for video in j_results['items']:

```

```

        complete_dict['channel_name'] =
video['snippet']['channelTitle']
        complete_video = {}
        complete_video['video_id'] =
video['id']['videoId']
        returned_statistics = {}
        returned_statistics =
crawlstatistics(video['id']['videoId'])
        complete_video['video_title'] =
returned_statistics[complete_video['video_id']]['vid
eo_title']
        complete_video['video_date'] =
returned_statistics[complete_video['video_id']]['vid
eo_date']
        statistics =
returned_statistics[complete_video['video_id']]['vid
eo_statistic']

complete_video['video_dislikeCount'] =
int(statistics['dislikeCount']) if 'dislikeCount' in
statistics else 0

complete_video['video_likeCount'] =
int(statistics['likeCount']) if 'likeCount' in
statistics else 0

complete_video['video_viewCount'] =
int(statistics['viewCount']) if 'viewCount' in
statistics else 0

complete_video['video_commentCount'] =
int(statistics['commentCount']) if 'commentCount' in
statistics else 0

        returned_comments = {}
        returned_comments =
crawlcomments(video['id']['videoId'])
        complete_video['video_comments']
= returned_comments

complete_dict['video'].append(complete_video)

```

```

        num_processed += 1
        if num_processed % 10 == 0:
            print("{} Video Processed:
{}").format
                                (num_processed,
datetime.datetime.now()))
        if 'nextPageToken' in j_results:
            parameters['pageToken'] =
j_results['nextPageToken']
        else:
            video_search_nextpage = False
            print("\nDone!\n{} Videos Processed in {}".format(
                num_processed, datetime.datetime.now() -
scrape_starttime))
            return complete_dict

```

Kode 5.2-13 menunjukkan fungsi untuk melakukan *crawl* video pada setiap akun Youtube pemda. Dimana Youtube memanggil channel id untuk mengambil setiap video milik akun tersebut.

Kode 5.2-7 Fungsi *Crawl* Setiap Pemda

```

def channellists():
    start_row = 1
    idpemda = spreadsheet.scrapingcolumn(3, start_row,
None)
    namapemda = spreadsheet.scrapingcolumn(1,
start_row, None)
    idchannelpemda = spreadsheet.scrapingcolumn(9,
start_row, -1)
    idchannelinfluencer =
spreadsheet.scrapingcolumn(12, start_row, -1)

    for satuanidpemda in idpemda[171:]:

        pemda_dict = {}
        pemda_dict['pemda_id'] = satuanidpemda
        pemda_dict['pemda_name'] =
namapemda[idpemda.index(satuanidpemda)]
        pemda_dict['resmi'] = {}
        pemda_dict['influencer'] = {}

        satuanidchannelpemda =
idchannelpemda[idpemda.index(satuanidpemda)]
        satuanidchannelinfluencer =

```

```

idchannelinfluencer[idpemda.index(satuanidpemda)]

    print("\n===== No.{:^2.5}
{:^5}".format(satuanidpemda,
pemda_dict['pemda_name']))
    if satuanidchannelpemda is not '':
        pemda_dict['resmi'] = crawlvideo(api_key,
satuanidchannelpemda, pemda_dict['pemda_id'],
pemda_dict['pemda_name'], 'resmi')
    elif satuanidchannelpemda is '':
        pemda_dict['resmi']['video'] = []

pemda_dict['resmi']['video'].append({'video_comments
': []})
    if satuanidchannelinfluencer is not '':
        pemda_dict['influencer'] = crawlvideo(api_key,
satuanidchannelinfluencer, pemda_dict['pemda_id'],
pemda_dict['pemda_name'], 'influencer')
    elif satuanidchannelinfluencer is '':
        pemda_dict['influencer']['video'] = []

pemda_dict['influencer']['video'].append({'video_com
ments': []})
    # Export JSON dalam bentuk string secara manual
    if satuanidpemda is idpemda[0]:
        exported = '[' + str(json.dumps(pemda_dict,
indent=4)) + ', '
        with open('Youtube.json', 'a') as file:
            file.write(exported)
    elif satuanidpemda is not idpemda[-1]:
        exported = str(json.dumps(pemda_dict,
indent=4)) + ', '
        with open('Youtube.json', 'a') as file:
            file.write(exported)
    else:
        exported = str(json.dumps(pemda_dict,
indent=4)) + ']'
        with open('Youtube.json', 'a') as file:
            file.write(exported)

```

Pada kode 5.2-14 akun milik pemda dimuat dalam bentuk *list* dimana data akun tersebut didapatkan dari pengambilan data

pada google spreadsheet. Rentang waktu mulai dan akhir diinisiasi pada kode ini. Selanjutnya *crawler* akan berjalan secara berulang sesuai dengan list yang berisi akun-akun pemda. Kode ini juga melakukan pengecekan apakah pemda memiliki akun atau tidak. Data hasil pada proses ini merupakan JSON lengkap yang berisi data dari keseluruhan akun pemda.

5.3 Pemilahan Data Crawling

Pada proses ini dilakukan pemilahan data antara data *crawling* yang bersifat lengkap dengan data yang dibutuhkan yaitu berupa data post pada setiap media sosial pemda. Hal ini dilakukan dengan menggunakan python, ekstensi **yaml** serta **pandas** untuk memilah data *crawl* yang berbentuk json. Berikut merupakan baris kode pada python.

Kode 5.3-1 Pemilahan Data

```
import json
import yaml
import pandas as pd
from pandas.io.json import json_normalize
with open('data.json') as f:
    d = json.load(f)
df = pd.io.json.json_normalize(data=d,
    record_path=[data yang diambil])
df = df.dropna(subset=[data yang diambil])
file = df[data yang disimpan]
file.to_csv('data.csv', sep=';', encoding='utf-8', index=False)
```

5.4 Labelling Data Post

Pada tahap ini dilakukan pemberian label pada data post yang telah dikumpulkan dan dipilah. Proses ini dilakukan secara manual dengan memanfaatkan fitur *data validation* pada MS. Excel.

Table 5.4-1 Hasil Labelling

Teks	Tipe
------	------

Sejarah Migas Indonesia Ternyata Bermula dari Kerajaan Sriwijaya http://t.co/oXiWzDlZBS #infoplg #infonasional	Edukasi Warga
HORE...PEMBAYARAN PBB DI LUTIM KINI SUDAH ONLINE	Informasi Layanan
Festival Lomba Bedug dan Gema Takbir Tingkat Kecamatan Pancoran - Sabtu 17 Juni 2017	Informasi Peristiwa
INDAHNYA WISATA BUKIT DAUN, BEKAS LAHAN TAMBANG https://t.co/pqG3gyXBhW lewat @tubankabgoid	Promosi Daerah
Bagaimana pendapat kawan kawan? https://t.co/PmfNZFHegN	Permintaan Informasi / Opini
maaf kemarin lagi maintenance https://t.co/UxHQ9pRUYA	Pemberitahuan Pemeliharaan

5.5 Pra-processing Data

Pada tahap ini dilakukan praprocessing data yang terdiri dari *case folding*, *data cleaning*, *stemming*, *tokenizing* dan *stopword removal*. Data yang didapatkan dibagi menjadi data train dan data test dengan proporsi 7:3. *Case folding* dilakukan untuk menyetarakan ejaan pada data sehingga menjadi *lowercase* atau *ppercase*. *Data cleaning* dilakukan untuk membersihkan data dari karakter-karakter yang tidak penting atau tidak bermakna

dengan menggunakan *regular expression (regex)*. *Data cleaning* dilakukan dengan menggunakan kode 5.5-2.

Selanjutnya dilakukan *stemming* dengan menggunakan library Sastrawi[14] melalui Python. *Stemming* berfungsi untuk

Kode 5.5-1 Data Cleaning dan Case Folding

```
df['text'] = df['text'].str.lower()
df['text'] = df['text'].replace('http://t.co/\w*',
'', regex=True)
df['text'] =
df['text'].replace('https?:\//\.*\//\w*', '',
regex=True)
df['text'] = df['text'].replace('www.*\//\w*', '',
regex=True)
df['text'] = df['text'].replace('\#\w*', '',
regex=True)
df['text'] = df['text'].replace('\@\w*', '',
regex=True)
df['text'] = df['text'].replace('[' +
string.punctuation + ']+', ' ', regex=True)
df['text'] = df['text'].replace('0|[1-9][0-9]*', '
', regex=True)
df['text'] =
df['text'].str.replace('[^\w\s#@/:%.,_-]', '',
flags=re.UNICODE)
df['text'] = df['text'].replace('\n', ' ',
regex=True)
df['text'] = df['text'].replace('\s+\s+', ' ',
regex=True)
```

menjadikan kata-kata menjadi kata dasar. *Stemming* dilakukan dengan menggunakan kode 5.5-2.

Kode 5.5-2 Stemming

```
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
df['text'] = df['text'].apply(stemmer.stem)
```

Tokenizing dilakukan untuk menjadikan rangkaian kalimat menjadi token (per kata). *Tokenizing* dilakukan dengan menggunakan library pyspark ml yaitu *regex tokenizer*. Kode 5.5-3 merupakan kode untuk melakukan *tokenizing*.

Kode 5.5-3 Tokenizing

```
# regular expression tokenizer
regexTokenizer = RegexTokenizer(inputCol="text",
outputCol="tokenized", pattern="\W")
```

Dan yang terakhir dilakukan *stopword removal* untuk menghilangkan kata-kata tidak penting atau sering muncul tapi tidak memiliki makna. *Stopword removal* dilakukan dengan menggunakan library pyspark ml yaitu *StopWordRemover* dan daftar *stopwords* berdasarkan penelitian oleh Fadhillah[15]. Kode 5.5-4 merupakan kode yang digunakan untuk melakukan *stopword removal*.

Kode 5.5-4 Stopword Removal

```
fadilah = open("id-stopwords.txt", "r")
stopwords = fadilah.read().split('\n')
stopwordsRemover =
StopWordsRemover(inputCol="tokenized",
outputCol="stopped").setStopWords(stopwords)
```

Dari tahap praprocessing akan menghasilkan data seperti berikut.

Table 5.5-1 Hasil Cleaning

Teks	Hasil Cleaning
#JasMerah Refleksi bagi kita semua, perlunya mengenalkan sejarah bagi generasi muda... https://t.co/EBnOV5rFtr	refleksi bagi kita semua perlu kenal sejarah bagi generasi muda

Table 5.5-2 Hasil Tokenizing

Hasil Cleaning	Tokenized
----------------	-----------

refleksi bagi kita semua perlu kenal sejarah bagi generasi muda	['refleksi', 'bagi', 'kita', 'semua', 'perlu', 'kenal', 'sejarah', 'bagi', 'generasi', 'muda']
---	---

Table 5.5-3 Hasil Stopword Remover

Tokenized	Stopword
['refleksi', 'bagi', 'kita', 'semua', 'perlu', 'kenal', 'sejarah', 'bagi', 'generasi', 'muda']	['refleksi', 'kenal', 'sejarah', 'generasi', 'muda']

5.6 Pembuatan dan Pengujian Model

Pada tahap ini dilakukan pembuatan model berdasarkan data yang telah dilakukan pre-processing. Hasil dari pembuatan model ini akan dijadikan *pipeline* yang selanjutnya dijadikan acuan untuk dilakukannya klasifikasi. Data yang telah dilakukan praprocessing selanjutnya dibagi menjadi 7:3 antara data training dan data testing. Pembagian data dilakukan dengan menggunakan kode 5.6-1.

Kode 5.6-1 Membagi Dataset

```
data = sqlContext.createDataFrame(df)
(trainingData, testData) = data.randomSplit([0.7,
0.3], seed = 100)
```

Selanjutnya data teks dari dataset diubah menjadi bentuk *vector*. Hal ini dilakukan agar diketahui pola teks dikarenakan mesin hanya bisa membaca angka. Proses ini menggunakan library *pyspark ml* yaitu *CountVectorizer*. Kode 5.6-2 merupakan kode yang melakukan proses tersebut.

Kode 5.6-2 Mengubah Teks Menjadi Vector

```
countVectors = CountVectorizer(inputCol="stopped",
                                outputCol="features", vocabSize=<vocab_size value>,
                                minDF=<minDF value>)
```

Selanjutnya teks label diubah menjadi indeks angka sehingga mempermudah mesin melakukan pengecekan label. Proses tersebut dilakukan dengan menggunakan library pypspark ml yaitu *StringIndexer* pada kode 5.6-3.

Kode 5.6-3 Merubah String Label menjadi Indeks

```
label_stringIdx = StringIndexer(inputCol = "tipe",
                                 outputCol = "label")
labels = label_stringIdx.fit(data).labels
```

Kemudian stages *tokenizing*, *stopwords removal*, *countVectors* dan *label_stringIdx* dijadikan satu untuk membentuk stages awal *pipeline*. Hal tersebut dilakukan dengan menggunakan kode 5.6-4.

Kode 5.6-4 Stages Awal Pipeline

```
early_stages = [regexTokenizer, stopwordsRemover,
                 countVectors, label_stringIdx]
```

Selanjutnya *model* dibuat berdasarkan *LinearSVC* dan *OneVSRest* untuk melakukan multiclass clasification yang terdapat pada library *machine learning* pada pypspark. Pada *LinearSVC* terdapat parameter *maxIter* dan juga *regParam*. Proses tersebut dilakukan dengan menggunakan kode 5.6-5.

Kode 5.6-5 Pembuatan Model SVM

```
lsvc = LinearSVC(labelCol="text", \
                  featuresCol="tipe", \
                  maxIter=10, \
                  regParam=0.03)
ovr = OneVsRest(classifier=lsvc)
```

Kemudian indeks label hasil prediksi dikembalikan menjadi indeks dalam bentuk string. Hal ini dilakukan untuk mempermudah membaca data pada database. Proses ini merupakan akhir dari *stages pipeline* dan selanjutnya disatukan

dengan proses sebelumnya menjadi satu *pipeline* dengan menggunakan kode 5.6-6

Kode 5.6-6 Perubahan Menjadi String dan Menyatukan Pipeline

```
#mengubah indeks prediksi menjadi string
labelConverter = IndexToString(inputCol="prediction",
outputCol="class", labels=labels)
#menyatukan pipeline
pipeline = Pipeline(stages= early_stages + [ovr,
labelConverter])
```

Selanjutnya dilakukan metode *fit* untuk mempelajari parameter model dengan menggunakan data training, dan metode *transform* untuk menerapkan pada data testing. Selanjutnya model yang disimpan setelah didapatkan hasil yang paling baik untuk digunakan pada saat proses klasifikasi data *stream*. Hal tersebut dilakukan dengan menggunakan kode 5.6-7.

Kode 5.6-7 Membuat Prediksi dan Menyimpan Model

```
#membuat prediksi dan melatih model
svmPipeline = pipeline.fit(trainingData)
predictions = svmPipeline.transform(testData)
#save pipeline
svmPipeline.save('svmPipeline')
```

Model yang dibuat diuji dengan menggunakan evaluator model yang dimiliki oleh library pyspark. Pengujian dilakukan beberapa kali untuk menemukan model dengan parameter yang menghasilkan hasil yang paling tinggi.

Kode 5.6-8 Mengubah ke dalam bentuk RDD Spark

```
pred_columns = ['type', 'tokenized', 'stopped',
'rawPrediction', 'probability', 'features', 'class']
pred_eval = predictions.select([column for column in
predictions.columns if column not in pred_columns])
rdd = pred_eval.rdd
```

Kode 5.6-8 digunakan untuk mengubah data menjadi bentuk rdd Spark sehingga dapat dilakukan pengujian.

Kode 5.6-9 Pengujian Model

```
#===== Testing Evaluation Start=====
```

Kode 5.6-2 Fungsi Pengujian Tiap Kategori

```
from pyspark.mllib.evaluation import
MulticlassMetrics
from pyspark.mllib.evaluation import
RegressionMetrics
evaluator =
MulticlassClassificationEvaluator(predictionCol="p
rediction")
evaluator.evaluate(predictions)
# Instantiate metrics object
metrics = MulticlassMetrics(predictionAndLabels)
# Overall statistics
precision = metrics.precision()
recall = metrics.recall()
f1Score = metrics.fMeasure()
accuracy = metrics.accuracy
print("Summary Stats")
print("Precision = %s" % precision)
print("Recall = %s" % recall)
print("F1 Score = %s" % f1Score)
print("Accuracy = %s" % accuracy)
labels = rdd.map(lambda lp:
lp.label).distinct().collect()
for label in sorted(labels):
    print("Class %s precision = %s" % (label,
metrics.precision(label)))
    print("Class %s recall = %s" % (label,
metrics.recall(label)))
    print("Class %s F1 Measure = %s" % (label,
metrics.fMeasure(label, beta=1.0)))
```

Kode 5.6-9 digunakan untuk melakukan pengujian atau validasi model pada beberapa parameter yang berbeda-beda. Selanjutnya dilakukan pengujian untuk tiap kelas kategori. Pengujian dilakukan dengan menggunakan kode berikut;

Berikut merupakan hasil dari pengujian tersebut;

```
labels = rdd.map(lambda lp:
lp.label).distinct().collect()
for label in sorted(labels):
    print("Class %s precision = %s" % (label,
metrics.precision(label)))
    print("Class %s recall = %s" % (label,
metrics.recall(label)))
    print("Class %s F1 Measure = %s" % (label,
metrics.fMeasure(label, beta=1.0)))
)
#===== Testing Evaluation End =====
```

Table 5.6-1 Nilai Per Kategori

Type	Precision	Recal	F-Measure
Promosi Daerah	0.736677	0.88015	0.802048
Informasi Peristiwa	0.710526	0.81	0.757009
Informasi Layanan	0.86645	0.869281	0.867863
Pemberitahuan Pemeliharaan	0.822981	0.765896	0.793413
Edukasi Warga	0.83121	0.805556	0.818182
Permintaan Informasi Opini	0.775591	0.625397	0.692443

5.7 Streaming Data Post

Pada tahap ini dilakukan pembuatan kode untuk melakukan *crawling* pada proses *stream*. Data yang masuk melalui didapatkan melalui Kafka dilakukan penguraian (*parsing*) untuk menyusun data agar sesuai dengan struktur dokumen yang akan disimpan dalam database. Proses *stream* data pada Facebook dan Youtube dilakukan dengan melakukan perulangan sistem *crawler* yang sama seperti pada sub bab 5.2 untuk masing-masing media sosial sedangkan untuk Twitter berbeda. Sebelumnya dilakukan inisiasi kafka untuk membuat topik untuk melakukan *crawling*.

Inisiasi ini dilakukan dengan menyalakan server zookeeper dan melakukan pengaturan terlebih dahulu.

Kode 5.7-1 Start server zookeeper

```

${KAFKA_HOME}/bin/zookeeper-server-start.sh
${KAFKA_HOME}/config/zookeeper.properties &

while ! nc -z localhost 2181; do
    echo "zookeeper activated !"
    sleep 20

    ${KAFKA_HOME}/bin/kafka-server-start.sh
    ${KAFKA_HOME}/config/server.properties &
    sleep 10 &
    echo "Kafka server activated"

```

Selanjutnya pembuatan topik yang dibutuhkan

Kode 5.7-2 Pembuatan topik

```

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic facebook-kafka-post

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic facebook-kafka-comment

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic twitter-kafka-post

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic twitter-kafka-comment

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic youtube-kafka-post

${KAFKA_HOME}/bin/kafka-topics.sh --create --
zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic youtube-kafka-comment

${KAFKA_HOME}/bin/kafka-topics.sh --list --
zookeeper localhost:2181

```


Selanjutnya inisiasi produser dan konsumer untuk tiap topik, inisiasi ini dilakukan pada file `egovbench_kafka.py`.

Kode 5.7-3 Inisiasi Produser dan Konsumer

```
class FacebookKafkaPost(Producer, Consumer):
    def __init__(self, topic='facebook-kafka-
post'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())

class FacebookKafkaComment(Producer, Consumer):
    def __init__(self, topic='facebook-kafka-
comment'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())

class TwitterKafkaPost(Producer, Consumer):
    def __init__(self, topic='twitter-kafka-
post'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())

class TwitterKafkaComment(Producer, Consumer):
    def __init__(self, topic='twitter-kafka-
comment'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())
```

```

class YoutubeKafkaPost(Producer, Consumer):
    def __init__(self, topic='youtube-kafka-
post'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())

class YoutubeKafkaComment(Producer, Consumer):
    def __init__(self, topic='youtube-kafka-
comment'):
        logging.info('[EGOVBENCH_KAFKA]>' + 'topic
connected: ' + topic)
        Producer.__init__(self, topic,
prompter=topic.upper())
        Consumer.__init__(self, topic,
prompter=topic.upper())

```

Proses *crawling* dilakukan secara berulang-ulang sesuai dengan keseluruhan akun pemda yang ada.

Kode 5.7-4 Parser Post

```

import logging
class Parser():
    def __init__(self, accountTypeKey,
accountIDKey, postIDKey, commentIDKey):
        self.account_type_key = accountTypeKey
        self.account_id_key = accountIDKey
        self.post_id_key = postIDKey
        self.comment_id_key = commentIDKey

```

```

    def getPostDocument(self, complete_dict):
        postdocument = complete_dict['post']
        postdocument['_id'] =
        postdocument[self.post_id_key]
        postdocument.pop(self.post_id_key, None)
        postdocument[self.account_id_key] =
        complete_dict['account'][self.account_id_key].lower()
        self.prompt('{{: {} ({}: {})) Post
document created!'.format(
            self.account_id_key,
            postdocument[self.account_id_key],
            self.post_id_key,
            postdocument['_id']
        ))
        return postdocument

class TwitterParser(Parser):
    """
    Subclass dari kelas Parser() untuk Twitter
    """
    def __init__(self):
        super().__init__('account_type',
        'account_id', 'tweet_id', 'reply_id')
class YoutubeParser(Parser):
    """
    Subclass dari kelas Parser() untuk Youtube
    """
    def __init__(self):
        super().__init__('channel_type',
        'channel_id', 'video_id', 'comment_id')
class FacebookParser(Parser):
    """
    Subclass dari kelas Parser() untuk
    Facebook
    """
    def __init__(self):
        super().__init__('page_type', 'page_id',
        'post_id', 'comment_id')

```

Kode 5.7-1 menunjukkan proses parsing pada setiap post media sosial dimana dalam class Parser terdapat fungsi `getPostDocument` untuk memarsing data menjadi sesuai

dengan kebutuhan pada pengerjaan tugas akhir ini. Selanjutnya terdapat 3 subclass yaitu class TwitterParser, YoutubeParser dan FacebookParser yang masing-masing subclass tersebut mendefinisikan data apa yang akan diparsing.

Kode 5.7-5 Mengirimkan Data ke Kafka

```
def send_message(self, dict_or_json):

    producer = self.topic.get_producer(
        serializer=self.serializer_method,
        sync=True,
        max_request_size=1024 * 1024 * 10,
        use_rdkafka=True,
        delivery_reports=False,
    )
    with producer as producer:
        try:
            producer.produce(dict_or_json)
        except Exception as e:
            logging.warning(e)
        self.promptProducer('(_id: {}) Data
sent!'.format(dict_or_json['_id']))
        latest_available_offsets =
json.dumps(self.topic.latest_available_offsets())
        logging.debug('Latest available offsets: '
+ latest_available_offsets)
```

Kode 5.7-2 berisi fungsi yang digunakan untuk mengirimkan data post Facebook dan Youtube ke kafka yang sebelumnya telah diparsing dimana data berupa dictionary atau JSON.

Kode 5.7-6 Proses Data dari Crawler

```
def pushPostDocument(self, complete_dict):
    post_document =
self.fp.getPostDocument(complete_dict)
    self.fmc.updatePost(post_document)
    self.fkp.send_message(post_document)
```

Kode 5.7-3 berisi fungsi pushDocument dimana berguna untuk memanggil method yang berfungsi untuk mengupdate post Facebook dan Youtube pada database MongoDB. Kode ini membutuhkan input dictionary atau JSON post.

Untuk proses *streaming* Twitter digunakan dengan menggunakan API yang disediakan untuk mengambil data tweet Twitter. Namun untuk menyalurkan data tetap menggunakan Kafka sebagai data *pipeline*. Proses ini langsung mengambil data akun-akun pemerintah melalui database secara langsung. Data tweet akan disalurkan ke dalam salah satu topik kafka.

Kode 5.7-7 Twitter Steam Listener

```
class StdOutListener(StreamListener):
    def on_data(self, data):
        producer.send_messages("Twitter-stream-
post", data.encode('utf-8'))
        print(data)
        return True
    def on_error(self, status):
        print(status)
```

Kode 5.7-4 menunjukkan kode yang digunakan sebagai penerus data ke dalam kafka. Pada kode ini juga dilakukan pengecekan terhadap data error yang masuk.

Kode 5.7-8 Ambil JSON

```
def ambil_list():
    list_accountID =
collection.distinct('account_id_number')
    return list_accountID
```

Kode 5.7-5 menunjukkan kode yang digunakan untuk mengambil daftar akun Twitter yang terdapat pada database. Akun yang digunakan hanya akun resmi milik pemda.

Kode 5.7-9 Streaming Tweet Twitter

```
access_token = "<var acces token>"
access_token_secret = "<var acces token secret>"
consumer_key = "<var consumer key>"
consumer_secret = "<var consumer secret>"
#Mongo Client
client = pymongo.MongoClient()
database = client['egovbench']
collection = database['Twitter_accounts']
```

```

if __name__ == '__main__':
    #Setup kafka client
    kafka = KafkaClient("localhost:9092")
    #Kafka simple producer
    producer = SimpleProducer(kafka)
    l = StdOutListener()
    #Twitter auth
    auth = OAuthHandler(consumer_key,
consumer_secret)
    auth.set_access_token(access_token,
access_token_secret)
    #Create stream api
    stream = Stream(auth, l,
wait_on_rate_limit=True,
wait_on_rate_limit_notify=True)

    while True:
        try:
            print("Stream started")

stream.filter(follow=['264821845'])
        except Exception as e:
            print("Stream stopped")
            print("type error: " + str(e))

```

Kode 5.7-6 merupakan kode yang berfungsi untuk melakukan stream tweet Twitter. Akun keseluruhan pemda dimuat dalam bentuk *list* dimana didapatkan dari fungsi *ambil_list*. Pada kode ini *client* dihubungkan dengan API Twitter dengan menggunakan *OAuthHandler* yang dibantu dengan library *tweepy*. Proses *streaming* ini dilakukan terus menerus dengan adanya pengecekan terhadap error. Apabila terjadi error maka akan dilakukan pengambilan data lagi. Data disimpan dalam bentuk JSON dari setiap tweet yang masuk.

5.8 Streaming Spark

Tahapan ini dilakukan setelah data *stream* masuk melalui kafka. Pengaturan konfigurasi Apache Spark lebih lengkapnya dapat dilihat pada penelitian M. Fikry Hazmi[16]. Data dari kafka kemudian ditarik berdasarkan topik stream masing-masing media sosial. Selanjutnya proses dilakukan terhadap data yang masuk. Proses dijalankan dengan cara perbatch dengan jangka waktu 10 detik sekali.

Kode 5.8-1 Inisiasi Spark Streaming Post

```
#List pemda
lists = ambil_list()

#Parsing Tweet
Twitter_tweet =
TwitterParsedPost.map(parsing_TwitterTweet)
#Parsing Facebook Post
Facebook_post =
FacebookParsedPost.map(parsing_FacebookPost)
#Parsing Youtube Post
Youtube_video =
YoutubeParsedPost.map(parsing_YoutubePost)

#Process Tweet
df_Twitter =
Twitter_tweet.foreachRDD(processTwitterTweet)
#Process Facebook Post
df_Facebook =
Facebook_post.foreachRDD(processFacebookPost)
#Process Youtube Video
df_Youtube =
Youtube_video.foreachRDD(processYoutubePost)
```

Kode 5.8-1 diinisiasi pemanggilan akun Twitter pemda dan proses data yang masuk dari setiap media sosial. Inisiasi akun hanya dilakukan pada Twitter saja karena proses *streaming* yang berbeda. Setelah melakukan *map* pada setiap data kemudian dilakukan proses pada setiap *rdd*.

Kode 5.8-2 Parsing Data Facebook & Youtube

```
def parsing_FacebookPost(post):
    post_id = post['_id']
    page_id = post['page_id']
    text = post['post_message']
    return post_id, text

def parsing_YoutubePost(post):
    video_id = post['_id']
    channel_id = post['channel_id']
    text = post['video_title']
    return video_id, text
```

Kode 5.8-2 merupakan kode yang berfungsi untuk parsing pada media sosial Facebook dan Youtube. Parsing dilakukan dengan masukan satu post.

Kode 5.8-2 Parsing Data Twitter

```
def parsing_TwitterTweet(tweet):
    tweet_id = ''
    tweet_createdDate = ''
    account_id = ''
    text = ''
    if 'text' in tweet:
        if 'RT @' not in tweet['text'] :
            if not tweet['truncated']:
                text = tweet['text']
            elif tweet['truncated']:
                text =
tweet['extended_tweet']['full_text']
                tweet_id = tweet['id_str']
                tweet_createdDate =
eu.formatTwitterTime(tweet['created_at'])
                account_id =
tweet['user']['screen_name'].lower()
    return tweet_id, tweet_createdDate,
    account_id, text
```

Kode 5.8-3 merupakan kode yang berfungsi untuk parsing pada data tweet Twitter. Kode ini berbeda karena pada Twitter dibutuhkan adanya mode extended tweet agar teks tidak terpotong.

Kode 5.8-4 Transform ke Pandas atau Spark Dataframe

```
def pd_to_sparkDF(df):
    df = spark.createDataFrame(df)
    return df

def sparkDF_to_pd(df):
    df = df.toPandas()
    df = df[df.text != '']
    return df
```

Pada kode 5.8-4 terdapat fungsi yang digunakan untuk melakukan transformasi dari pandas menjadi spark dataframe maupun sebaliknya. Hal ini dibutuhkan pada proses RDD setiap media sosial.

Kode 5.8-5 Proses RDD Pada Setiap Media Sosial

```
def processTwitterTweet(rdd):
    if rdd.isEmpty():
        print('No Data Received from Twitter\n')
    else:
        schema = StructType( [
            StructField('_id', StringType()),
            StructField('tweet_createdDate',
StringType()),
            StructField('account_id',
StringType()),
            StructField('text', StringType()),
        ])
        df = spark.createDataFrame(rdd, schema)
        df = sparkDF_to_pd(df)

        process_item_post(df)

def processFacebookPost(rdd):
    if rdd.isEmpty():
        print('No Data Received from Facebook\n')
    else:
        schema = StructType( [
            StructField('_id', StringType()),
            StructField('page_id', StringType()),
            StructField('text', StringType()),
        ])
        df = spark.createDataFrame(rdd, schema)
        df = sparkDF_to_pd(df)
```

```

        process_item_post(df)

def processYoutubePost(rdd):
    if rdd.isEmpty():
        print('No Data Received from Youtube\n')
    else:
        schema = StructType( [
            StructField('_id', StringType()),
            StructField('channel_id',
StringType()),
            StructField('text', StringType()),
        ])
        df = spark.createDataFrame(rdd, schema)
        df = sparkDF_to_pd(df)

        process_item_post(df)

```

Pada kode 5.8-5 dilakukan proses RDD pada setiap media sosial dimana proses ini melakukan pengecekan rdd serta melakukan pembuatan skema spark dataframe dari data rdd yang telah berhasil diparsing dari setiap media sosial.

5.9 Klasifikasi Data

Pada tahap ini dilakukan preprocessing sekaligus klasifikasi data *streaming* melalui Apache Spark. Preprocessing dilakukan dengan mengambil data *streaming* yang masuk melalui Kafka. File yang dibutuhkan diakses melalui folder apps pada aplikasi egovbench. Preprocessing dan klasifikasi dilakukan dengan menggunakan kode berikut.

Kode 5.9-1 Fungsi Preprocessing

```

def preprocess(df): # Pandas
    # case folding dan data cleaning
    df['text'] = df['text'].str.lower() #casefolding
    df['text'] = df['text'].apply(lambda x:
re.sub(r'https?:\/\/\.*\/\w*', '', x)) #remove
hyperlink
    df['text'] = df['text'].apply(lambda x:
re.sub(r'www.*\/\w*', '', x)) #remove URL
    df['text'] = df['text'].apply(lambda x:
re.sub(r'#\w+', '', x)) # remove hashtag
    df['text'] = df['text'].apply(lambda x:

```

```

re.sub(r'@\w+', '', x)) # remove mention
df['text'] = df['text'].apply(lambda x: "
".join(re.findall('[\w]+',x))) # remove punctuation
df['text'] = df['text'].apply(lambda x:
re.sub(r'0|[1-9][0-9]*', '', x)) # remove number
df['text'] = df['text'].apply(lambda x:
re.sub(r'\n', ' ', x)) # remove
df['text'] = df['text'].apply(lambda x:
re.sub(r'\s+\s+', ' ', x)) # remove multiple space

# stemming
df['text'] = df['text'].apply(stemmer.stem)

return df

```

Kode 5.9-1 merupakan kode yang berfungsi untuk melakukan preprocessing data namun pada kode ini tidak dilakukan *tokenizing* dan *stopword removal*. Pemuatan model SVM dilakukan untuk mengklasifikasikan data yang didapat, model yang dimuat merupakan model yang telah dibuat sebelumnya. Hasil model ini diletakkan pada folder apps pada aplikasi streaming. Model dimuat dengan kode berikut.

Kode 5.9-2 Load SVM Model

```
svmPipeline = PipelineModel.load('svmPipeline')
```

Pada kode 5.9-3 dilakukan pengecekan terhadap data yang telah didapatkan apakah data tersebut mengandung nilai kosong atau data sudah tidak dibutuhkan. Data yang sudah tidak dibutuhkan selanjutnya dibuang dengan menggunakan fungsi garbage collection (gc).

Kode 5.9-3 Pengecekan Data

```

def process_item_post(df):
    df = preprocess(df)
    if df.empty:
        if 'account_id' in df:
            print('No Data Received from Twitter\n')
        elif 'page_id' in df:
            print('No Data Received from Facebook\n')
        elif 'channel_id' in df:
            print('No Data Received from Youtube\n')
    gc.collect()

```

```

else:
    df = pd_to_sparkDF(df)
    df = svm_pipeline(df)
    json_result = json_parsing(df)
    insert_mongo_post(json_result)
    gc.collect()

```

Pada kode 5.9-4 dilakukan proses pengklasifikasian untuk setiap data yang telah siap atau sudah melalui tahap preprocessing. Proses ini membutuhkan file `egovbench_sparkstream_post.py` yang diakses melalui folder `apps` pada aplikasi `egovbench`.

Kode 5.9-4 Proses Klasifikasi Stream

```

def svm_pipeline(dataset): #Another preprocess
(tokenizing, stopping) & classification (svm)
    predict = svmPipeline.transform(dataset)
    drop_list = ['tokenized', 'stopped', 'features',
'rawPrediction', 'probability', 'prediction']
    data = predict.select([column for column in
predict.columns if column not in drop_list])
    df = sparkDF_to_pd(data)
    if 'account_id' in df.columns:
        df.rename(columns={'text': 'tweet_message'},
inplace=True)
    elif 'page_id' in df.columns:
        df.rename(columns={'text': 'post_message'},
inplace=True)
    elif 'channel_id' in df.columns:
        df.rename(columns={'text': 'video_message'},
inplace=True)
    return df

```

Selanjutnya data yang telah terklasifikasi dimasukkan ke dalam database MongoDB masing-masing media sosial. Untuk konfigurasi MongoDB selengkapnya dapat dilihat pada penelitian Dyaksa Hanindito[17]. File yang dibutuhkan adalah `egovbench_mongo.py` yang diakses melalui folder `apps` pada aplikasi `egovbench`. Selanjutnya inisiasi tersebut dilakukan dengan menggunakan kode berikut.

Kode 5.9-6 Inisiasi MongoDB

```

#Mongo Client
client = pymongo.MongoClient()
database = client['egovbench']
collection_pemda = database['listpemda']
collection_dinas = database['dinas']
collection_TwitterPost =
database['Twitter_posts']
collection_FacebookPost =
database['Facebook_posts']
collection_YoutubePost =
database['Youtube_posts']

```

Pada kode 5.9-5 dilakukan inisiasi database pada masing-masing media sosial. Kemudian dilakukan parsing data hasil ke dalam bentuk JSON sehingga mempermudah memasukkan data ke dalam database. Proses parsing ini membutuhkan file `egovbench_parsing.py` yang dapat diakses pada folder `apps` pada aplikasi `egovbench`. Parsing data dilakukan dengan kode berikut.

Kode 5.9-5 Parsing Pandas ke JSON

```

def json_parsing(df): # Pandas
    json_result = df.to_json(orient='records')
    json_result = json.loads(json_result)

    return json_result

```

Selanjutnya dijalankan fungsi insert untuk memasukkan data yang terklasifikasi ke dalam database sesuai dengan masing-masing pemda. Fungsi insert dilakukan melalui `egovbench-mongo.py` yang diakses melalui folder `apps` pada aplikasi `egovbench`. Hal tersebut dilakukan dengan kode berikut.

Kode 5.9-7 Insert Hasil Klasifikasi ke Database

```

def insert_mongo_post(json_result):
    Twitter_num_processed = 0
    Facebook_num_processed = 0
    Youtube_num_processed = 0
    if 'account_id' in json_result[0]:
        for tweet in json_result:
            twMongo.updatePost(tweet)
            Twitter_num_processed += 1
        print("{}          Twitter          Tweet
Processed\n".format(Twitter_num_processed))
    elif 'page_id' in json_result[0]:
        for post in json_result:
            fbMongo.updatePost(post)
            facenum_processed += 1
        print("{}          Facebook          Post
Processed\n".format(Facebook_num_processed))
    elif 'channel_id' in json_result[0]:
        for video in json_result:
            ytMongo.updatePost(video)
            num_processed += 1

```

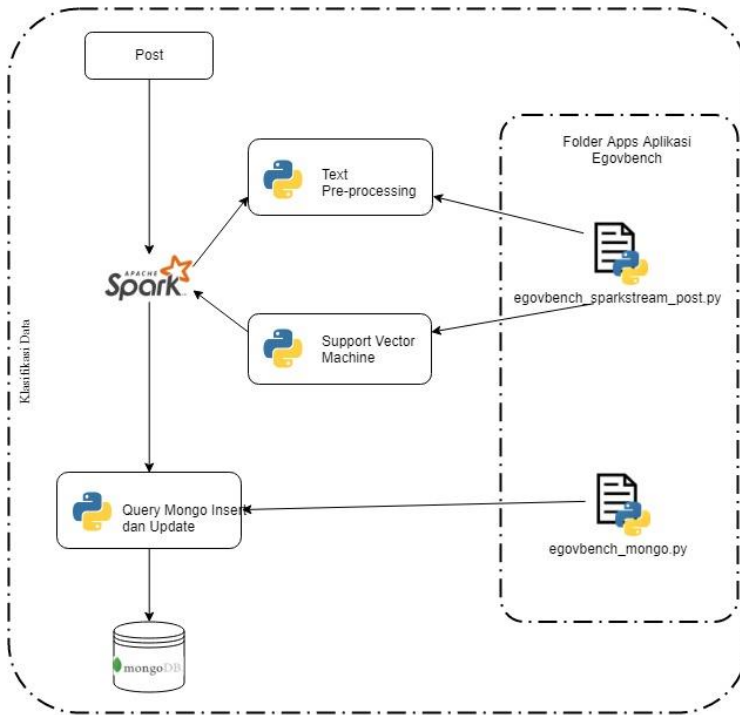
Pada kode 5.9-8 dilakukan pembaruan pada data post dengan query *upsert* pada MongoDB. Query ini sekaligus melakukan pengecekan untuk melihat apakah data terduplikasi atau tidak apabila daa sudah ada maka akan dilakukan *update* sedangkan bila tidak maka akan dilakuakn *insert*. Fungsi ini dilakukan melalui *egovbench_mongo.py* yang diakses melalui folder *apps* pada aplikasi *egovbench*

Kode 5.9-8 Upsert Query pada MongoDB

```

def updatePost(self, postDocument):
    """
        Meng-update post pada post collection,
        dengan filter _id (yang merupakan atribut post_id_key
        dari postDocument).
        Bila filter tidak terpenuhi, akan
        menambah document baru ke dalam post collection.
    """
    self.prompt('{{: {}} Updating post please
wait      .      .      {}'.format(self.post_id_key,
postDocument['_id']))
    try:
        self.post_collection.update_one(
            {
                '_id': postDocument['_id']
            },
            {
                '$set': postDocument
            },
            upsert=True
        )
        self.prompt('{{: {}} Post updated
!'.format(self.post_id_key, postDocument['_id']))
    except Exception as e:
        logging.warning(e)

```



Gambar 5.9-1 Klasifikasi Data Stream

Pada gambar 5.9-1 merupakan alur proses klasifikasi data dimana post yang masuk diproses oleh Apache Spark, selanjutnya dilakukan praproses dan klasifikasi data dengan menggunakan script `egovbench_sparkstream_post.py`. Pada proses klasifikasi SVM, model klasifikasi diambil dari hasil pembuatan sebelumnya dimana diakses melalui folder `svmModel` pada folder `apps` pada aplikasi `egovbench`. Setelah dilakukan klasifikasi dilakukan query insert serta update untuk memasukkan data hasil klasifikasi ke dalam database MongoDB dengan menggunakan script `egovbench_mongo.py`.

5.10 Visualisasi

Tahap ini merupakan pembuatan visualisasi grafik serta statistik harian yang dihasilkan berdasarkan data post pada media sosial pemda yang telah diproses melalui *streaming*. Untuk konfigurasi visualisasi lebih lengkapnya dapat dilihat pada penelitian Paranody Risky P[18]. Terdapat beberapa grafik yang digunakan pada tugas akhir ini yaitu diagram batang untuk menampilkan data label klasifikasi dan untuk memperlihatkan nilai standar deviasi post, diagram garis digunakan untuk menampilkan tren atau perkembangan post media sosial pemda, diagram pie digunakan untuk memperlihatkan rasio tipe post pada media sosial Facebook dan Twitter milik pemda.

Kode 5.10-1 Query Harian Post

```

date_default_timezone_set("Asia/Jakarta");
$pemda = listPemdaModel::all()->count();
$Facebook_resmi =
listPemdaModel::where('Facebook_resmi', '!=', '')-
>count();
$Facebook_influencer =
listPemdaModel::where('Facebook_influencer', '!=',
'')->count();
$Twitter_resmi =
listPemdaModel::where('Twitter_resmi', '!=', '')-
>count();
$Twitter_influencer =
listPemdaModel::where('Twitter_influencer', '!=',
'')->count();
$Youtube_resmi =
listPemdaModel::where('Youtube_resmi', '!=', '')-
>count();
$Youtube_influencer =
listPemdaModel::where('Youtube_influencer', '!=',
'')->count();
$post =
FacebookPostsModel::where('post_createdDate',
date("Y-m-d"))->count()
+
TwitterPostsModel::where('tweet_createdDate',
date("Y-m-d"))->count()
+
YoutubePostsModel::where('video_createdDate',
date("Y-m-d"))->count();

```

Visualisasi ini dilakukan dengan menggunakan library Highcharts.

```

$postKlasifikasi      =      $post      -
FacebookPostsModel::where('post_message',      '')-
>where('post_createdDate', date("Y-m-d"))->count() +
TwitterPostsModel::where('tweet_message',      '')-
>where('tweet_createdDate', date("Y-m-d"))->count() +
YoutubePostsModel::where('video_title',      '')-
>where('video_createdDate', date("Y-m-d"))->count();

```

Kode 5.10-1 berfungsi untuk menampilkan statistik harian pos yang masuk dan terklasifikasi. Pada query ini juga ditampilkan jumlah akun media sosial pemda resmi maupun influencer yang terdaftar. Selanjutnya untuk visualisasi untuk statistik harian dapat dilihat pada gambar berikut.

Daily Stats

Tanggal Hari Ini

2018-06-25

Total Pemerintah Daerah	548
Total Facebook Resmi / Influencer	148 / 155
Total Twitter Resmi / Influencer	201 / 175
Total Youtube Resmi / Influencer	113 / 59
Total Post Masuk	0
Total Komentar Masuk	0
Total Post Terklasifikasi	0
Total Komentar Terklasifikasi	0
Total Komentar Terkategorisasi	6
Total Komentar Tidak Terkategori	1
Total Komentar Terduplikasi	1
Rata-Rata Komentar per Pemda	0.01%

Gambar 5.10-1 Visualisasi Statistik Harian

Kode 5.10-2 Query Visualisasi Klasifikasi Post

```

    $pemda = listPemdaModel::where('_id',
(int)$id)->first();
    $labelKlasifikasiPost = ['Edukasi Warga',
'Informasi Layanan', 'Informasi Peristiwa',
'Permintaan Informasi Opini', 'Promosi Daerah',
'Pemberitahuan Pemeliharaan'];
    //Facebook resmi
    $postFacebookResmi =
FacebookPostsModel::where('page_id', $pemda-
>Facebook_resmi)->get();

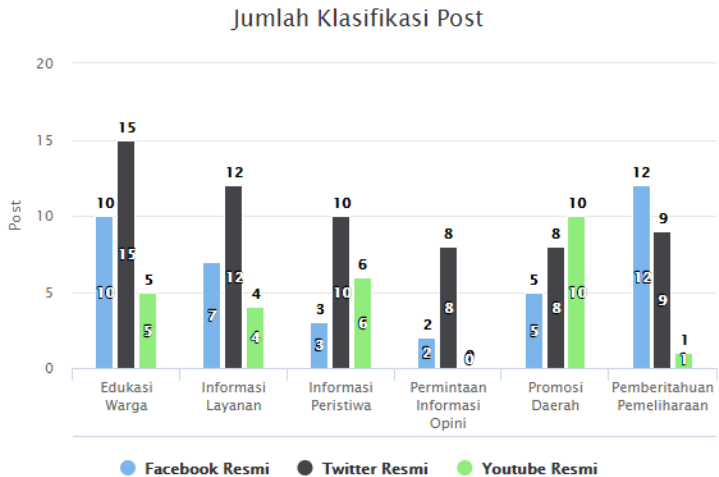
    foreach($labelKlasifikasiPost as $lk) {
        $jumlahPostFacebookResmi[] =
$postFacebookResmi->where('class', $lk)->count();
    }
    //Twitter resmi
    $tweetTwitterResmi =
Twitter_replyModel::where('account_id', $pemda-
>Twitter_resmi)->get();

    foreach($labelKlasifikasiPost as $lk) {
        $jumlahTweetTwitterResmi[] =
$tweetTwitterResmi->where('class', $lk)->count();
    }
    //Youtube resmi
    $postYoutubeResmi =
YoutubeCommentsModel::where('channel_id', $pemda-
>Youtube_resmi)->get();

    foreach($labelKlasifikasiPost as $lk) {
        $jumlahPostYoutubeResmi[] =
$postYoutubeResmi->where('class', $lk)->count();
    }

```

Kode 5.10-2 berfungsi untuk melakukan query hasil klasifikasi post pada tiap media sosial yang diambil dari database MongoDB untuk mendapatkan nilai total post yang terklasifikasi tiap label. Selanjutnya data tersebut ditampilkan dengan menggunakan gambar berikut.

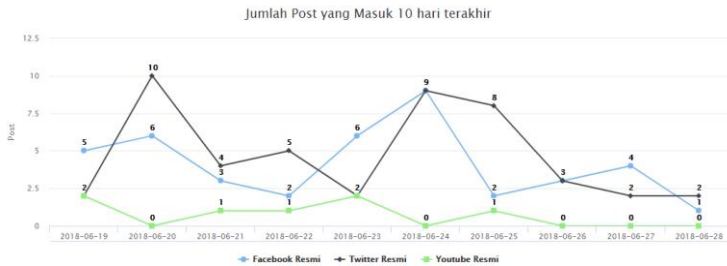


Gambar 5.10-2 Klasifikasi Post

Kode 5.10-3 digunakan untuk melakukan query 10 hari terakhir, untuk melihat pergerakan dan perkembangan post pada setiap pemda. Selanjutnya ditampilkan dengan menggunakan kode berikut.

Kode 5.10-3 Query Pergerakan Post

```
foreach ($tanggal as $tgl) {
    $hitungPostFBResmi[] =
    FacebookPostsModel::where('post_createdDate', $tgl)-
    >where('page_id', $pemda->Facebook_resmi)->count();
    $hitungPostTWResmi[] =
    TwitterPostsModel::where('tweet_createdDate', $tgl)-
    >where('account_id', $pemda->Twitter_resmi)-
    >count();
    $hitungPostYTResmi[] =
    YoutubePostsModel::where('video_createdDate', $tgl)-
    >where('channel_id', $pemda->Youtube_resmi)-
    >count();
}
```



Gambar 5.10-3 Pergerakan Post

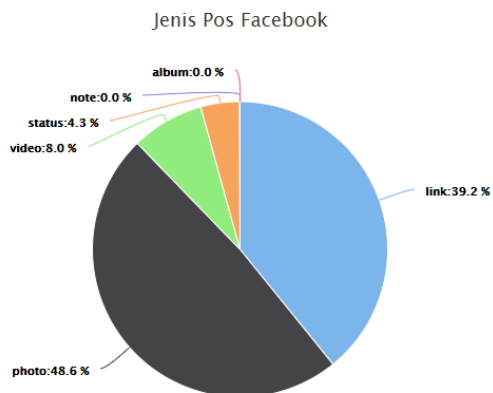
Kode 5.10-4 digunakan untuk melakukan query untuk mengambil tipe post media sosial Facebook dan Twitter dari database MongoDB. Selanjutnya ditampilkan dengan menggunakan kode berikut.

Kode 5.10-4 Query Tipe Post Facebook & Twitter

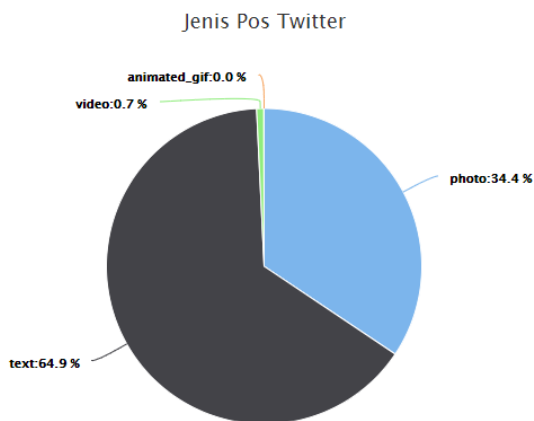
```
//Facebook resmi
$namaPostTypeFacebook = ['link', 'photo',
'video', 'status', 'note', 'album'];

foreach($namaPostTypeFacebook as $nptf) {
    $jumlahTypePostFacebookResmi[] =
$postFacebookResmi->where('post_type', $nptf)-
>count();
}
//Twitter resmi
$namaTweetTypeTwitter = ['animated_gif',
'video', 'text', 'photo'];

foreach($namaTweetTypeTwitter as $nttt) {
    $jumlahTypeTweetTwitterResmi[] =
$tweetTwitterResmi->where('tweet_type', $nttt)-
>count();
}
```



Gambar 5.10-4 Jenis Pos Facebook



Gambar 5.10-5 Jenis Pos Twitter

Bab VI

HASIL DAN PEMBAHASAN

6.1 Analisa Hasil Pemodelan

Pada Analisa hasil permodelan akan membahas mengenai hasil pengujian model menggunakan *support vector machine*. Adapun langkah – langkah yang dilakukan dijelaskan pada hal berikut.

6.1.1 Pemuatan Data

Total data yang digunakan dari hasil crawling yang digunakan berjumlah 6197 data. Adapun detail data yang digunakan dibagi berdasarkan sosial media ditunjukkan dengan Tabel 6.1.1.

Table 6.1-1 Pembagian Data Berdasarkan Sosial Media

Media Sosial	Jumlah
Facebook	919
Twitter	3357
Youtube	1921

Data yang telah didapatkan dilakukan pelabelan secara manual dengan menggunakan bantuan Microsoft Excel berikut merupakan rincian data hasil pelabelan.

Table 6.1-2 Hasil Pelabelan Data

Kategori	Facebook	Twitter	Youtube	Total
Edukasi Warga	43	430	547	1020
Informasi Layanan	143	600	289	1032
Informasi Peristiwa	44	506	496	1046
Promosi Daerah	296	199	561	1056
Permintaan Informasi Opini	52	945	17	1014

Pemberitahuan Pemeliharaan	341	677	11	1029
Total	998	3275	1921	6197

Data yang telah terlabel selanjutnya dilakukan preprocessing dimana terdapat beberapa tahapan yaitu casefolding, cleaning, stemming, tokenizing dan stopwords removal. Pada data awal terdapat 158,919 total kata dari 6197 data post. Terdapat perubahan data setelah melewati beberapa tahapan yang ada pada preprocessing dimana jumlah keseluruhan menjadi berubah.

Table 6.1-3 Hasil Tahapan Preprocessing

Tahapan	Jumlah Kata
Data Awal	158919
<i>Data Cleaning</i>	142000
<i>Casefolding</i>	142000
<i>Stemming</i>	142000
<i>Tokenizing</i>	142000
<i>Stopword Removal</i>	100711

Pada tahapan *data cleaning* terjadi pengurangan jumlah kata sebanyak 16919 kata karena pada tahapan ini terjadi pembersihan dari karakter dan angka yang tidak diperlukan. Pada tahapan *casefolding* sampai *tokenizing* tidak terjadi perubahan jumlah kata dikarenakan pada tahap tersebut hanya dilakukan penyetaraan ejaan, merubah menjadi kata dasar dan membuat token dari data kata yang ada. Pada tahap *stopword removal* terjadi pengurangan jumlah kata hal ini disebabkan terjadinya penghapusan kata-kata yang dianggap tidak penting atau tidak bermakna.

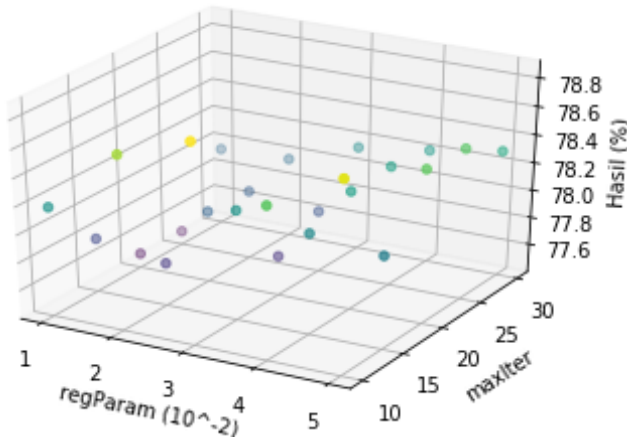
6.1.2 Pembentukan Model Klasifikasi dan Validasi

Dalam melakukan pembentukan model SVM, model malui tahap validasi atau pengujian dengan cara merubah variabel pada `maxIter` dan `regParam` yang dimiliki oleh library *machine learning* (ml) pada Apache Spark syang kemudian dinilai dengan metode *precision*, *recall*, dan F-Measure. Penentuan `maxIter` berada pada nilai antara 10 hingga 30 dengan loncatan 5 setiap pergantian nilai. Sedangkan untuk `regParam` berurutan

mulai dari 0.01 sampai 0.05. Hasil tersebut diuji dengan menggunakan fungsi penilaian yang dimiliki oleh Apache Spark.

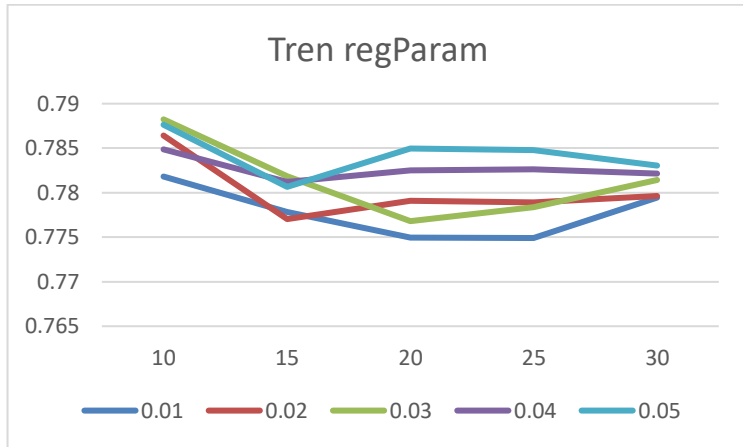
Table 6.1-4 Hasil Evaluasi Presisi, Recall, F-Measure

Parameter		regParam				
		0.01	0.02	0.03	0.04	0.05
maxIter	10	0.7818	0.7864	0.7882	0.7849	0.7876
	15	0.7778	0.777	0.7819	0.7812	0.7807
	20	0.775	0.7791	0.7768	0.7825	0.785
	25	0.7749	0.7789	0.7784	0.7826	0.7848
	30	0.7795	0.7797	0.7814	0.7821	0.783



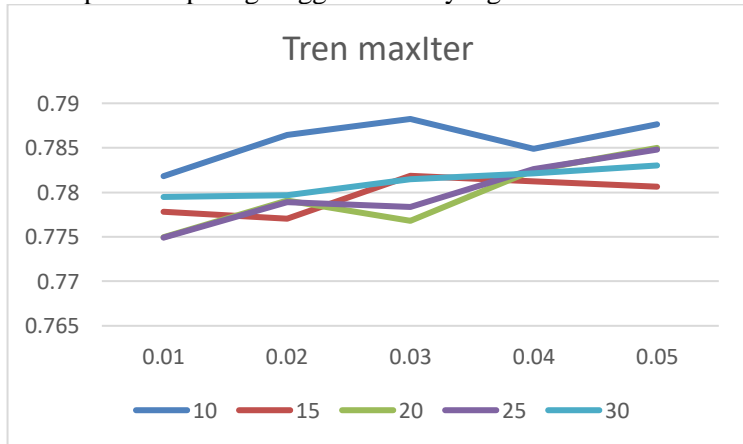
Gambar 6.1-1 Hasil Evaluasi

Berdasarkan hasil validasi yang dilakukan dapat diketahui bahwa semakin tinggi nilai regParam dimulai dari nilai 0.04 maka akan menghasilkan nilai yang berada pada batas kurang lebih 78% pada setiap percobaan maxIter yang dilakukan. Namun walaupun nilai rata-rata lebih rendah dibandingkan regParam 0.04 dan 0.05, nilai maksimal hanya terdapat pada regParam 0.03.



Gambar 6.1-2 Trend regParam

Sedangkan pada parameter maxIter dapat dilihat bahwa nilai maxIter 10 selalu mendapat hasil diatas dari semua parameter yang digunakan pada pengujian ini. MaxIter 10 mampu mencapai nilai paling tinggi diantara yang lain diatas 78.5%.



Gambar 6.1-3 Trend maxIter

Sehingga maxIter dengan nilai 10 dan regParam dengan nilai 0.03 dipilih karena memiliki nilai yang paling tinggi yaitu 78.82% dengan rincian nilai setiap media sosial sebagai berikut.

Table 6.1-5 Hasil Tiap Media Sosial

Media Sosial	Nilai
Facebook	0.7581
Twitter	0.8323
Youtube	0.7701

Pada data tersebut Twitter mendapat nilai yang paling tinggi disebabkan penggunaan data yang lebih banyak dari Facebook dan Youtube, sebanyak 3278 data. Sedangkan Facebook mendapat nilai terendah dari masing-masing media sosial karena data yang digunakan lebih rendah yaitu hanya 998 data. Selanjutnya dilakukan perhitungan *precision*, *recall* dan *f-measure* pada masing-masing label hingga didapatkan hasil berikut.

Table 6.1-6 Hasil Evaluasi Keseluruhan

Tipe	Precision	Recal	F-Measure
Promosi Daerah	73.67%	88.02%	80.20%
Informasi Peristiwa	71.05%	81.00%	75.70%
Informasi Layanan	86.65%	86.93%	86.79%
Pemberitahuan Pemeliharaan	82.30%	76.59%	79.34%
Edukasi Warga	83.12%	80.56%	81.82%
Permintaan Informasi Opini	77.56%	62.54%	69.24%
AVG/TOTAL	79.06%	79.27%	78.85%

Berdasarkan hasil pelabelan data sebelumnya terdapat kecenderungan kata yang mengacu pada label tertentu dimana kumpulan kata tersebut dapat dilihat pada tabel berikut.

Table 6.1-7 Kata Kunci Setiap Label

Kategori Konten	Kata Yang Memungkinkan
Edukasi Warga	Penyuluhan, Sosialisasi, Edukasi, himbauan, warisan
Informasi Peristiwa	Kegiatan, upacara, gotong royong, pembinaan

Promosi Daerah	Profil, amazing, wonderful, wisata, pesona, sensasi, wow, menjelajah, yuk, jelajah, ayo, eksotisme
Informasi Layanan	Buka, layanan, pelayanan, resmikan, peluncuran, sistem, launching, online, luncurkan, terapkan
Permintaan Informasi / Opini Kepada Warga	Bagaimana, menurut, dimata, tanggapan
Pemberitahuan Pemeliharaan (<i>maintenance</i>)	Maaf, maintenance, terkendala, offline, gangguan

Dari hasil tersebut didapatkan nilai *precision* sebesar 79,06% dan *recall* sebesar 79,27%. Dari hasil tersebut dapat diambil kesimpulan bahwa performa efektifitas klasifikasi model dapat dikatakan cukup baik dikarenakan nilai *precision* dan *recall* cukup seimbang. Nilai tersebut menunjukkan hasil yang lebih baik dibandingkan dengan hasil percobaan lainnya. Tetapi terdapat satu label kategori yang memiliki nilai *recall* dibawah 70% dimana hal ini jauh dari yang diharapkan. Hal ini bisa terjadi karena data yang digunakan kurang merepresentasikan label tersebut. Hasil dari proses pemodelan ini selanjutnya dipakai untuk melakukan klasifikasi post pada proses *streaming*.

6.2 Analisa Data

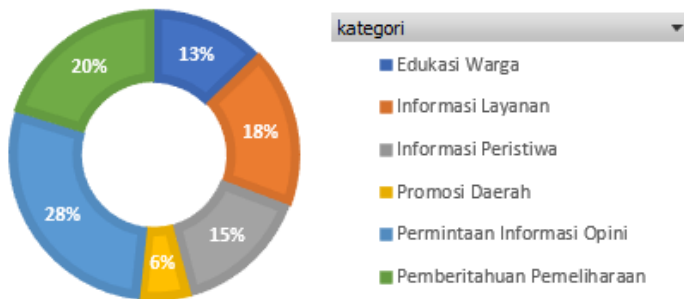
Berdasarkan data yang telah dikumpulkan, data Facebook dan Twitter dibagi berdasarkan tipe yang telah ditentukan. Pada data Twitter diketahui terdapat 3358 data, yang diantaranya terdapat 3355 data bertipe text dan 2 lainnya bertipe photo. Pada tipe text dapat diketahui bahwa sebesar 28% dari total data bertipe text masuk kedalam kategor Permintaan Opini Informasi dan rasio paling kecil sebesar 6% berada pada kategori Promosi Daerah. Sedangkan untuk tipe Photo hanya terdapat pada 2 kategori yaitu Permintaan Informasi Opini dan Promosi Daerah yang

dimana masing-masing hanya bernilai 1 data. Untuk detail data pada tipe text dapat dilihat pada tabel dan gambar berikut.

Table 6.2-1 Persebaran Data Twitter Tipe Text

Kategori	Jumlah
Edukasi Warga	430
Informasi Layanan	600
Informasi Peristiwa	506
Promosi Daerah	198
Permintaan Informasi Opini	944
Pemberitahuan Pemeliharaan	677
Grand Total	3355

PERSERBARAN KATEGORI TWITTER TIPE TEXT



Gambar 6.2-1 Persebaran Data Twitter Tipe Text

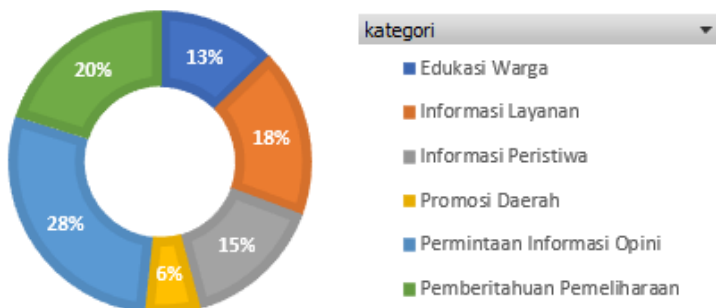
Untuk keseluruhan persebaran data Twitter pada setiap kategori dapat dilihat pada tabel dan gambar berikut.

Table 6.2-2 Persebaran Data Twitter Tiap Kategori

Kategori	Jumlah
Edukasi Warga	430
Informasi Layanan	600
Informasi Peristiwa	506
Promosi Daerah	199

Permintaan Informasi Opini	945
Pemberitahuan Pemeliharaan	677
Grand Total	3357

PERSEBARAN KATEGORI TWITTER



Gambar 6.2-2 Persebaran Twitter Tiap Kategori

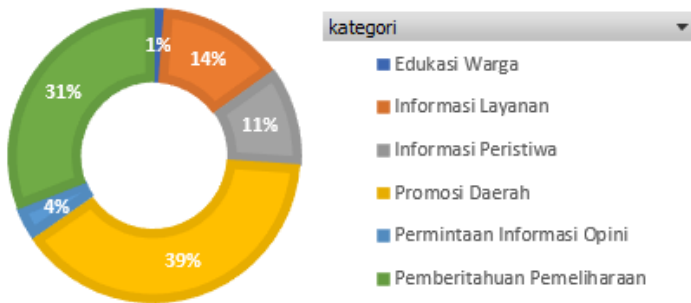
Berdasarkan data yang dihasilkan hanya sebesar 6% saja yang berada pada area kategori Promosi Daerah.

Pada data Facebook diketahui terdapat 998 data yang dimana kemudian dibagi berdasarkan tipe. Terdapat 4 tipe yang tercatat yaitu link, photo, status dan video. Pada tipe link sebesar 39% didominasi oleh kategori Promosi Daerah dan paling kecil sebesar 1% berada pada kategori Edukasi Warga. Berikut detail dari persebaran data pada tipe link.

Table 6.2-3 Persebaran Data FB Tipe Link

Kategori	Jumlah
Edukasi Warga	2
Informasi Layanan	23
Informasi Peristiwa	18
Promosi Daerah	65
Permintaan Informasi Opini	6
Pemberitahuan Pemeliharaan	51
Grand Total	165

PERSEBARAN KATEGORI FACEBOOK TIPE LINK



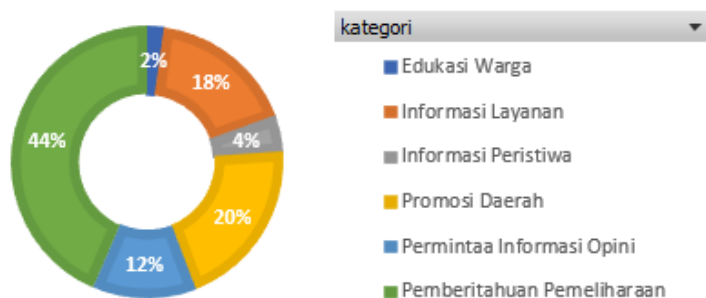
Gambar 6.2-3 Persebaran Data FB Tipe Link

Pada tipe photo sebesar 44% didominasi oleh kategori Pemberitahuan Pemeliharaan dan paling kecil sebesar 2% berada pada kategori Edukasi Warga. Berikut detail dari persebaran data pada tipe photo. Berikut detail dari persebaran data pada tipe photo.

Table 6.2-4 Persebaran Data FB Tipe Photo

Kategori	Jumlah
Edukasi Warga	3
Informasi Layanan	24
Informasi Peristiwa	6
Promosi Daerah	28
Permintaa Informasi Opini	17
Pemberitahuan Pemeliharaan	60
Grand Total	138

PERSEBARAN KATEGORI FACEBOOK TIPE PHOTO



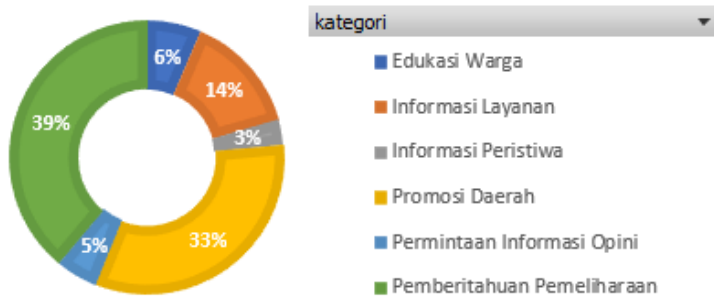
Gambar 6.2-4 Persebaran Data FB Tipe Photo

Pada tipe status sebesar 39% didominasi oleh kategori Pemberitahuan Pemeliharaan dan paling kecil sebesar 3% berada pada kategori Informasi Peristiwa. Berikut detail dari persebaran data pada tipe status. Berikut detail dari persebaran data pada tipe status.

Table 6.2-5 Persebaran Data FB Tipe Status

Kategori	Jumlah
Edukasi Warga	38
Informasi Layanan	84
Informasi Peristiwa	17
Promosi Daerah	193
Permintaan Informasi Opini	29
Pemberitahuan Pemeliharaan	230
Grand Total	591

PERSEBERAN KATEGORI FACEBOOK TIPE STATUS



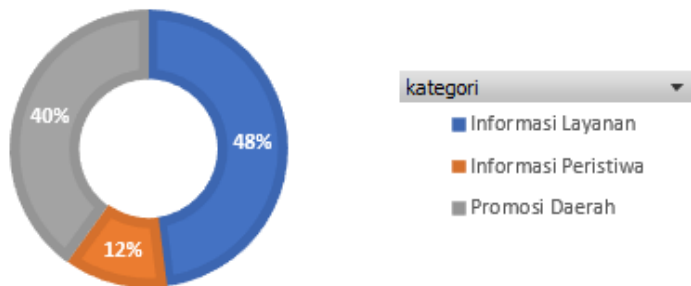
Gambar 6.2-5 Persebaran Data FB Tipe Status

Pada tipe video sebesar 48% didominasi oleh kategori Informasi Layanan dan paling kecil sebesar 12% berada pada kategori Informasi Peristiwa. Berikut detail dari persebaran data pada tipe video. Berikut detail dari persebaran data pada tipe video.

Table 6.2-6 Persebaran Data FB Tipe Video

Kategori	Jumlah
Informasi Layanan	12
Informasi Peristiwa	3
Promosi Daerah	10
Grand Total	25

PERSEBARAN KATEGORI FACEBOOK TIPE VIDEO



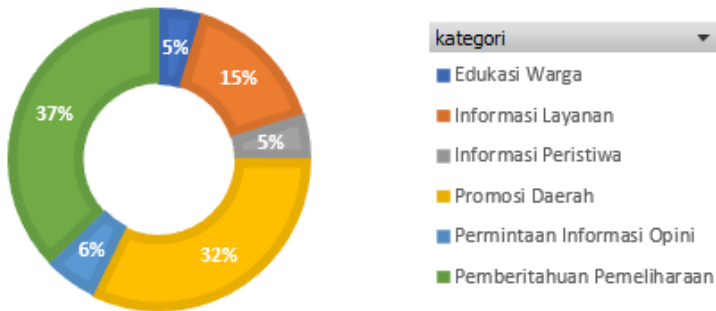
Gambar 6.2-6 Persebaran Data FB Tipe Video

Untuk keseluruhan persebaran data Facebook pada setiap kategori dapat dilihat pada tabel dan gambar berikut.

Table 6.2-7 Persebaran Data FB Tiap Kategori

Kategori	Jumlah
Edukasi Warga	43
Informasi Layanan	143
Informasi Peristiwa	44
Promosi Daerah	296
Permintaan Informasi Opini	52
Pemberitahuan Pemeliharaan	341
Grand Total	919

PERSEBARAN KATEGORI FACEBOOK



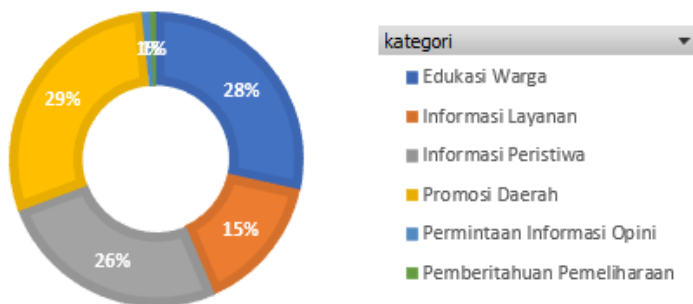
Gambar 6.2-7 Persebaran Data FB Tiap Kategori

Pada data Youtube diketahui terdapat 1921 dimana tipe data hanya bersifat tunggal yaitu video saja. Untuk persebaran data sebesar 29% didominasi oleh kategori Promosi Daerah sedangkan sekitar 1% berada pada kategori Permintaan Informasi Opini dan Pemberitahuan Pemeliharaan. Berikut detail persebaran dari data Youtube.

Table 6.2-8 Persebaran Data Youtube Tiap Kategori

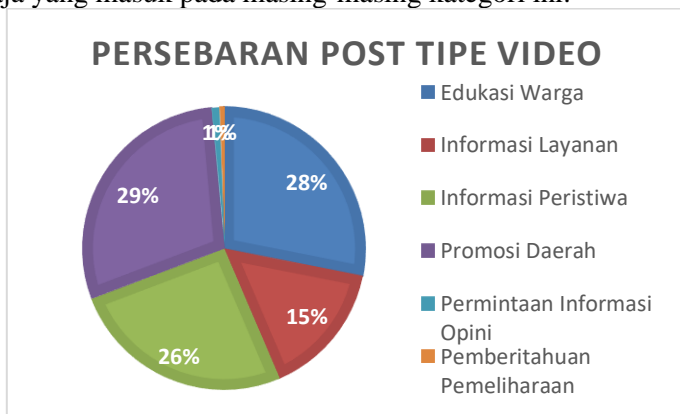
Kategori	Jumlah
Edukasi Warga	547
Informasi Layanan	289
Informasi Peristiwa	496
Promosi Daerah	561
Permintaan Informasi Opini	17
Pemberitahuan Pemeliharaan	11
Grand Total	1921

PERSEBARAN KATEGORI YOUTUBE



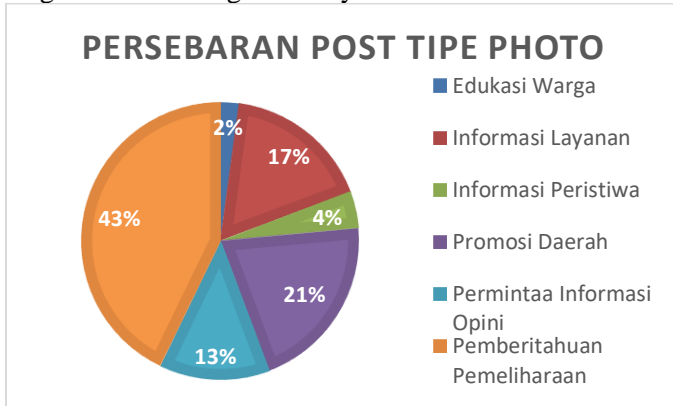
Gambar 6.2-8 Persebaran Data Youtube Tiap Kategori

Dari keseluruhan data diatas dapat dibedakan tipe data secara umum menjadi 3 bagian. Pertama adalah tipe **teks** yang terdiri dari Twitter tipe text dan Facebook tipe status dan link. Kedua adalah tipe **photo** yang terdiri dari Twitter tipe photo dan Facebook tipe photo. Dan terakhir adalah tipe **video** yang terdiri dari data Youtube dan Facebook tipe video. Berdasarkan pembagian 3 tipe data tersebut diketahui bahwa post berupa video sangat jarang sekali termasuk ke dalam kategori Pemberitahuan Pemeliharaan dan Permintaan Informasi Opini terlihat dari keseluruhan data bertipe video hanya sekitar 1 % saja yang masuk pada masing-masing kategori ini.



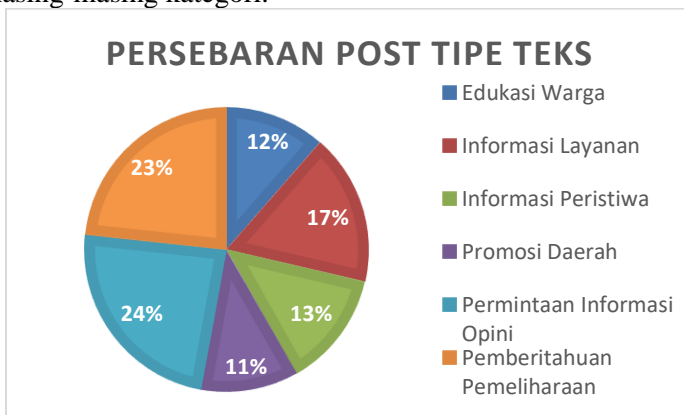
Gambar 6.2-9 Statistik Tipe Video

Pada tipe photo dapat diketahui bahwa pesebaran yang terjadi cukup menonjol dengan 43% data termasuk pada kategori Pemberitahuan Pemeliharaan, 2% masuk dalam kategori Edukasi Warga, 4% masuk dalam kategori Informasi Peristiwa sedangkan untuk kategori lainnya berkisar antara 13-21% data.



Gambar 6.2-10 Statistik Tipe Photo

Pada tipe teks persebaran data berkisar antara 10-24% untuk masing-masing kategori.



Gambar 6.2-11 Statistik Tipe Teks

Halaman ini sengaja dikosongkan

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan kesimpulan dan saran dalam pengerjaan tugas akhir.

7.1 Kesimpulan

Berdasarkan proses yang telah dilakukan dalam pengerjaan tugas akhir dengan judul “Rancang Bangun Aplikasi Untuk Klasifikasi Post Pada Sosial Media Pemerintah Daerah Di Indonesia Menggunakan Support Vector Machine (SVM)” yang telah dilakukan, dapat disimpulkan sebagai berikut:

1. Penelitian ini membuktikan bahwa metode crawling data dengan memanfaatkan Graph API Facebook, Twitter API dan Google Youtube API v3 mampu menjalankan proses pengumpulan data post milik akun media sosial Facebook, Twitter dan Youtube milik pemerintah daerah.
2. Hasil dari pengumpulan data dengan cara crawling data post dari 464 akun pemda resmi, diantaranya 149 akun dari media sosial Facebook, 202 akun dari media sosial Twitter dan 113 akun dari media sosial Youtube, didapatkan data sebanyak 226978 dengan rincian 73922 dari Facebook, 139211 dari Twitter dan 13845 dari Youtube.
3. Pemodelan *Support Vector Machine* dapat digunakan untuk melakukan klasifikasi post dengan menggunakan label tertentu sesuai dengan penelitian sebelumnya. Hasil terbaik didapatkan dengan menggunakan 6197 data dengan menggunakan parameter maxIter 10 dan regParam 0.03. Model tersebut menghasilkan nilai *presisi* dan *recall* sebesar 78,82% dengan rincian Twitter 82,32%, Facebook 75,81% dan Youtube 77,01%. Twitter memiliki nilai paling tinggi disebabkan karena data yang digunakan lebih banyak sebesar 3278 data dari total data yang digunakan 6197. Setiap pelabelan yang dilakukan sesuai dengan kecenderungan kata yang muncul pada setiap label

yang dapat dilihat pada tabel 6.1-7 Kata Kunci Tiap Label.

4. Data yang terkumpul dibagi menjadi 3 tipe data yaitu teks, photo dan video. Tipe teks memiliki persebaran 10-24% untuk setiap kategori. Tipe photo lebih condong ke arah Pemberitahuan Pemeliharaan sebesar 43% dan paling rendah pada kategori Edukasi Warga. Tipe video lebih condong ke arah Promosi Daerah sebesar 29% dan Edukasi Warga sebesar 28%, dan paling rendah berada pada kategori Pemberitahuan Pemeliharaan dan Permintaan Informasi Opini terlihat dari keseluruhan data bertipe video hanya sekitar 1 %.
5. Pada data *streaming* masih banyak ditemukan kesalahan pelabelan dikarenakan nilai yang didapat adalah sekitar 78% maka masih didapat banyak kesalahan pada klasifikasi data yang masuk secara *streaming*. Hal ini disebabkan karena salah satunya proses pelabelan yang dilakukn hanya mengacu pada satu persepektif saja sehingga ditemui banyak bias pada pelabelan data *streaming*.

7.2 Saran

Dalam pengerjaan tugas akhir ini masih terdapat kekurangan yang dapat diperbaiki. Sara diharapkan dapat membantu penelitian selanjutnya agar dapat lebih baik. Saran penulis untuk penelitian selanjutnya sebagai berikut :

1. Proporsi data didistribusikan secara merata menurut kebutuhan dan banyaknya label untuk klasifikasi.
2. Pembagian proporsi data training dan data testing yang berbeda untuk pembuatan model klasifikasi.
3. Percobaan pembuatan model klasifikasi dilakukan dengan menggunakan metode klasifikasi yang berbeda untuk melihat kemungkinan hasil yang lebih baik.
4. Untuk aplikasi Egovbench kedepannya perlu dlakukan penelitian lebih lanjut untuk pembuatan model klasifikasi yang dinamis sehingga model dapat melakukan pembaruan model secara otomatis dan

memperbanyak *insight* dalam aplikasi untuk meningkatkan analisa media sosial.

5. Proses pelabelan selanjutnya diharapkan dapat dilakukan dengan menggunakan lebih dari satu perspektif sehingga mengurangi bias pada pembuatan model klasifikasi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] We Are Social, "Digital in 2018 in Southeast Asia Part 2 - South-East," 18:15:38 UTC.
- [2] "PERMENPAN_NO_83_THN_2012_TENTANG-PEDOMAN-PEMANFAATAN-MEDIA-SOSIAL-INSTANSI-PEMERINTAH.pdf." .
- [3] N. Kurniasih, *Penggunaan Media Sosial bagi Humas di Lembaga Pemerintah*. 2013.
- [4] S. Hofmann, D. Beverungen, M. Räckers, and J. Becker, "What makes local governments' online communications successful? Insights from a multi-method analysis of Facebook," *Gov. Inf. Q.*, vol. 30, no. 4, pp. 387–396, Oct. 2013.
- [5] S. Alam and D. Walker, "The public Facebook: A case of Australian government Facebook pages and participation," in *Proceedings of ACIS*, 2011.
- [6] H. M. Abdelsalam, C. G. Reddick, S. Gamal, and A. Al-shaar, "Social media in Egyptian government websites: Presence, usage, and effectiveness," *Gov. Inf. Q.*, vol. 30, no. 4, pp. 406–416, Oct. 2013.
- [7] I. Dilrukshi, K. De Zoysa, and A. Caldera, "Twitter news classification using SVM," in *Computer Science & Education (ICCSE), 2013 8th International Conference on*, 2013, pp. 287–291.
- [8] M. Magnusson, P. Bellström, and C. Thoren, "Facebook usage in government—a case study of information content," 2012.
- [9] J. C. Bertot, P. T. Jaeger, and J. M. Grimes, "Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies," *Gov. Inf. Q.*, vol. 27, no. 3, pp. 264–271, Jul. 2010.

- [10] “Computer Glossary, Computer Terms - Technology Definitions and Cheat Sheets from WhatIs.com - The Tech Dictionary and IT Encyclopedia.” [Online]. Available: <http://whatis.techtarget.com/>. [Accessed: 26-Nov-2017].
- [11] “Indonesia Digital Landscape 2018.pdf.” .
- [12] S. R. Gunn, “Support vector machines for classification and regression,” *ISIS Tech. Rep.*, vol. 14, no. 1, pp. 5–16, 1998.
- [13] “Apache Kafka,” *Apache Kafka*. [Online]. Available: <https://kafka.apache.org/>. [Accessed: 26-Feb-2018].
- [14] “Sastrawi | PHP library untuk stemming Bahasa Indonesia | Stemmer Bahasa Indonesia.” [Online]. Available: <http://sastrawi.github.io/>. [Accessed: 25-Jun-2018].
- [15] F. Z. Tala, “A study of stemming effects on information retrieval in Bahasa Indonesia,” *Inst. Log. Lang. Comput. Univ. Van Amst. Neth.*, 2003.
- [16] M. F. HAZMI, “DEVELOPING AN APPLICATION FOR NETIZEN’S COMMENTS,” p. 145.
- [17] D. HANINDITO, “RANCANG BANGUN SISTEM PENILAIAN REAKSI MASYARAKAT TERHADAP AKUN SOSIAL MEDIA PEMERINTAH DAERAH,” p. 201.
- [18] N. R. PRATOMO, “RANCANG BANGUN APLIKASI UNTUK KATEGORISASI KOMENTAR NETIZEN PADA MEDIA SOSIAL PEMERINTAH DAERAH TERHADAP SKPD BERDASARKAN FREKUENSI KATA KUNCI SKPD,” p. 171 .

BIODATA PENULIS



Penulis lahir di kota pahlawan, Surabaya pada tanggal 13 September 1995. Walau lahir di kota pahlawan, penulis merupakan rakyat biasa. Anak ketiga dari empat bersaudara yang telah menempuh pendidikan formal dari SD hingga SMA. Pada tahun 2014, penulis melanjutkan pendidikan ke jenjang yang lebih tinggi, yaitu di Institut Teknologi Sepuluh Nopember (ITS) Surabaya, sebagai mahasiswa departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi (FTIK). Terdaftar sebagai pemilik NRP 05211440000153. Selama menjadi mahasiswa, penulis banyak mengikuti kegiatan kemahasiswaan, antara lain seminar, organisasi dan kajian. Penulis diberi tanggung jawab oleh organisasi BEM Fakultas Teknologi Informasi dan Komunikasi (BEM FTIK) sebagai Kepala Departemen Organizational Social Responsibility (OSR) pada tahun kepengurusan 2016/2017. Disamping aktif dalam kegiatan kemahasiswaan, penulis juga pernah menjadi asisten praktikum mata kuliah Desain Basis Data. Pada tahun ke-4, penulis tertarik dengan bidang Social Media Analysis, sehingga mengambil bidang minat laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui email gunturprakoso13@gmail.com